HOCHSCHULE MITTWEIDA
UNIVERSITY OF APPLIED SCIENCES

DIPLOMARBEIT

Herr
Stefan Thämmig

Untersuchung von Approximationsverfahren und Verfahren zur Validierung der Modellgüte von Ersatzmodellen

2010

Fakultät: **Mathematik/Naturwissenschaften/Informatik**

Diplomarbeit

Untersuchung von Approximationsverfahren und Verfahren zur Validierung der Modellgüte von Ersatzmodellen

Autor:
Stefan Thämmig

Studiengang:
Angewandte Mathematik

Seminargruppe:
MA06w1

Erstprüfer:
Prof. Dr. rer. nat. Thomas Villmann

Zweitprüfer:
Dipl. Math. Martin Stöcker

Mittweida, September 2010

Bibliographische Angaben

Thämmig, Stefan: Untersuchung von Approximationsmethoden und Verfahren zur Validierung der Modellgüte, 77 Seiten, 23 Abbildungen, 20 Tabellen, Anhang, Hochschule Mittweida (FH), Fakultät Mathematik/Naturwissenschaften/Informatik

Referat

Das Ziel der Diplomarbeit ist es, Strategien zur Ersatzmodellkonstruktion zur Approximationsbibliothek der EngineeringToolbox hinzuzufügen. Die Eignung verschiedener Approximationsverfahren für den Einsatz in der Bauteilsimulation ist zu prüfen und ausgewählte Verfahren sind zu implementieren. Die Güte der Modellierung soll mit bereits vorhandenen Approximationsmodellen verglichen werden. Dazu sind Methoden zur globalen Beurteilung der Qualität der Modelle aus der Literatur zu recherchieren und auf die Modelle anzuwenden. Zusätzlich sind Kriterien zu untersuchen, die die lokale Approximationsgenauigkeit eines Ersatzmodells bewerten können.
Im ersten Teil der Arbeit werden Approximationsmethoden vorgestellt, die für mehrdimensionale Daten angewendet werden können. Mithilfe der im zweiten Teil dargelegten Bewertungskriterien werden die untersuchten Verfahren hinsichtlich ihrer Approximationsgüte verglichen und Empfehlungen bezüglich ihres Einsatzes gegeben.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Symbolverzeichnis

α	Nachbarschaftsreichweite
$\mathbf{b}$	Vektor der rechten Seite mit den Stützwerten
b	Bestimmtheitsmaß
b_{Mod}	modifiziertes Bestimmtheitsmaß
β	Glättungsparameter der Shepard-Interpolation
$\mathbf{C}$	Korrelationsmatrix
$\mathbf{Corr}$	Korrelation zweier Punkte
$\mathbf{c}$	Vektor mit skalaren Koeffizienten der Ansatzfunktionen
c_j	skalare Koeffizienten der Ansatzfunktionen
D	Definitionsbereich
$\det(\mathbf{A})$	Determinante der Matrix $\mathbf{A}$
$diag$	Diagonalmatrix
d_N	Normalenabstand
Δ	absoluter Fehler
Δ_{Max}	maximaler absoluter Betragsfehler
Δ_{Mean}	mittlerer absoluter Betragsfehler
Δ_{MQ}	mittlerer quadratischer Fehler
Δ_{SQ}	summierter quadratischer Fehler
δ_{ij}	Kronecker-Delta
E_{NG}	Energiefunktion des Neural Gas
E_{RBF}	Energiefunktion von RBF-Netzen
ε	systematischer Fehler der Kriging-Interpolation
ε_{Max}	maximaler relativer Betragsfehler
ε_{Mean}	mittlerer relativer Betragsfehler
$\epsilon_{\mathbf{v}}$	Lernschrittweite für die Zentren

ϵ_λ	Lernschrittweite für Steilheit der Gaußglocken
$\epsilon_{\mathbf{\Omega}}$	Lernschrittweite für die Skalierungsmatrizen
ϵ_w	Lernschrittweite für Gewichte der Gaußglocken
$f(\mathbf{x}^*)$	geschätzter Funktionswert für eine Auswertungsstelle
$\mathfrak{G}$	Maximalgrad
$\mathbf{H}$	Hessematrix
$h_\alpha(k_j)$	Nachbarschaftsfunktion
$\mathbf{I}$	Einheitsmatrix
K	Anzahl der Testpunkte
k_j	Rang des j-ten Zentrums in RBF-Netzen
$\mathbf{L}$	Likelihood-Funktion
$L(x)$	LAGRANGE-Polynom
$\mathbf{\Lambda}_j$	symmetrische, positiv definite Matrizen für allgemeinere Metrik
λ^2	Steilheit der Gaußglocke
N	Anzahl der Stützstellen
$\mathbb{N}_0$	natürliche Zahlen einschließlich der Null
n	Anzahl der Ansatzfunktionen
M	Anzahl der Stützstellen innerhalb eines Radius R
m	Dimension des Datenraumes
μ	Mittelwert für die Kriging-Interpolation
$\mathbf{\Omega}_j$	Skalierungsmatrix der Daten in RBF-Netzen
$\mathfrak{O}$	Maximalordnung
$P_N(x)$	Polynom N-ten Grades
Φ	Matrix mit Werten der Ansatzfunktionen an den Stützstellen
$\varphi(x)$	eindimensionale Ansatzfunktion
$\phi(\mathbf{x})$	mehrdimensionale Ansatzfunktion
$\mathbf{Q}$	orthogonale Matrix
$\mathbf{R}$	obere Dreiecksmatrix
R	Radius
$\mathfrak{R}$	Maximalrang

σ^2	Varianz
$\mathbf{v}$	Zentren in RBF-Netzen
$\mathbf{W}$	Matrix zur Wichtung der Stützstellen
$w_{exp}\left(x^{(i)}\right)$	Wichtung einer Stützstelle mittels Gaußfunktion
$w_i\left(x\right)$	Wichtung bei der Shepard-Interpolation
x_i	Stützstelle
$\mathbf{x}^{(i)}$	Stützstellenvektor
$\mathbf{x}^*$	Auswertungsstelle
y_i	Stützwert zu einer Stützstelle
$y^{(i)}$	Stützwert zu einem Stützstellenvektor
$\hat{y}^{(i)}$	geschätzter Funktionswert
$\mathbf{Z}$	Matrix, die zeilenweise die Vektoren $\mathbf{z}$ enthält
$\mathbf{z}$	Vektor für Tensorprodukte
$\langle\cdot,\cdot\rangle$	Skalarprodukt
$\\|\cdot\\|$	Norm
$\\|\cdot\\|_2$	Euklidische Norm

Vorwort

In technischen Anwendungen spielt die Optimierung eine immer größere Rolle. Um komplexe Bauteile zu optimieren, werden verstärkt Methoden der mathematischen Optimierung genutzt. Da bei diesen komplexen Problemstellungen klassische Verfahren nur selten zufriedenstellende Ergebnisse liefern, werden stattdessen mathematische Ersatzmodelle genutzt, welche die aufwändigen Zielfunktionen nachbilden können. Diese Diplomarbeit stellt Verfahren vor, die mithilfe einer begrenzten Anzahl von Punkten Funktionen modellieren können. Das bestimmte Modell ermöglicht eine Analyse der unbekannten Funktionen hinsichtlich Linearität und Einfluss von verschiedenen Variablen auf die Zielfunktionswerte.
Um gewährleisten zu können, dass das mathematische Ersatzmodell die unbekannte Funktion ausreichend genau nachbildet, werden Kriterien untersucht, die Approximationsverfahren und damit die globale Güte der Ersatzmodelle bewerten. Zudem spielt für Optimierungsverfahren auch die lokalen Güte eine sehr entscheidende Rolle, weshalb auch Verfahren untersucht werden, welche die lokale Approximationsgenauigkeit abschätzen.

Die vorliegende Arbeit gliedert sich wie folgt: Kapitel 1 liefert die mathematischen Grundlagen der Approximation. Dabei werden zunächst die Begriffe Interpolation und Approximation definiert und deren Unterschiede dargelegt. Anschließend werden für beide Problemstellungen bekannte Verfahren für eindimensionale Daten vorgestellt. Im letzten Teil erfolgt die Überleitung zur Approximation mehrdimensionaler Daten. Das Kapitel 2 beschreibt Verfahren, die zur Approximation mehrdimensionaler Daten geeignet sind. Dabei wird zunächst die Approximation im quadratischen Mittel für eindimensionale Probleme auf mehrdimensionale verallgemeinert, sowie einige Varianten dieses Ansatzes untersucht. Im Weiteren werden die radialen Basisfunktionen, ein speziell für mehrdimensionale Daten verwendbares Verfahren, analysiert. Die Grundideen der in der Approximationsbibliothek bereits enthaltenen Verfahren Shepard-Interpolation und Krigingverfahren werden gegen Ende dieses Kapitels dargelegt. Dabei wird für das globale Verfahren von Shepard zudem ein lokaler Ansatz untersucht.
In Kapitel 3 werden Methoden zur Beurteilung der Güte von mathematischen Ersatzmodellen betrachtet. Dabei werden zunächst anwendbare globale Bewertungskriterien

dargelegt, anschließend nach einer Methode gesucht, um lokal, d.h. für einen bestimmten Datenpunkt den Fehler des Approximationsmodells zu schätzen.
Die Bewertung der betrachteten Approximationsverfahren anhand der in Kapitel 3 erläuterten globalen Gütekriterien erfolgt im Kapitel 4. Dabei wird die Leistungsfähigkeit der einzelnen Methoden an analytischen Testproblemen und einem Beispiel aus der Technik analysiert. Kapitel 5 fasst die gewonnen Erkenntnisse zusammen und liefert einen Ausblick auf mögliche Verbesserungen und neue Ansätze.

1 Mathematische Grundlagen

In diesem Kapitel werden die grundlegenden Definitionen für die folgenden Kapitel dargestellt. Der erste Abschnitt definiert die numerischen Methoden Interpolation und Approximation und verdeutlicht deren Unterschiede. In den folgenden beiden Teilen werden gebräuchliche Interpolations- und Approximationsverfahren für eindimensionale Daten vorgestellt. Der letzte Abschnitt dieses Kapitels verallgemeinert die Methoden auf mehrdimensionale Daten.

1.1 Interpolation und Approximation

Die Begriffe Interpolation und Approximation bezeichnen jeweils eine Klasse von Problemen und Verfahren aus der numerischen Mathematik. Das Hauptproblem der Interpolation ist die Bestimmung einer Funktion $f(x)$ zu gegebenen geordneten Paaren $(x_i, y_i)\,,\ i = 1, \ldots, N$, für die gilt $f(x_i) = y_i$.

Definition 1.1. Eine Funktion $f(x)$ heißt *Interpolationsfunktion* auf den Stützpunkten (x_i, y_i), wenn die Interpolationsbedingungen $f(x_i) = y_i,\ \forall i = 1, \ldots, N$ erfüllt sind.

Bei der Interpolation von Stützpunkten werden zumeist Linearkombinationen aus mehreren, sich nur durch Parameter unterscheidenden, Basisfunktionen φ_i verwendet. Die gewonnene Kurve kann dazu benutzt werden, die Funktion zwischen den Stützstellen abzuschätzen oder eine komplizierte Funktion durch eine leichter zu handhabende zu ersetzen. Die Stützstellen liegen auf der Kurve, zwischen ihnen kann es jedoch zu starken Abweichungen kommen.
Auch bei der Approximation wird das Ziel gesetzt, mit Hilfe von N Stützpunkten eine Funktion $f(x)$ zu finden. Im Gegensatz zur Interpolation wird jedoch nicht mehr verlangt, dass die gegebenen Punkte von der Funktion erreicht werden. Stattdessen wird deutlich mehr Wert auf die Glattheit der Kurve gelegt. Dieses Verfahren wird vor allem bei Messdaten, die größeren Ungenauigkeiten unterliegen, bevorzugt, weil dabei ein direktes Ansteuern der einzelnen Punkte zu starken Abweichungen zwischen den

Stützstellen führen könnte. Des Weiteren können für die Approximation deutlich weniger Ansatzfunktionen verwendet werden, während bei der Interpolation die Anzahl der Funktionen und Stützstellen übereinstimmen.

1.2 Interpolationsverfahren für eindimensionale Daten

1.2.1 Polynominterpolation

Zur Interpolation von eindimensionalen Daten, d. h. von geordneten Paaren reeller Zahlen $(x_i, y_i)\,,\ i = 0, \ldots, N$, werden sehr oft Polynome verwendet, da sich diese leicht integrieren und differenzieren lassen. Für $N+1$ Wertepaare wird das eindeutig bestimmte Polynom $P_N(x) = a_0 + a_1 \cdot x + \ldots + a_N \cdot x^N$ N-ten Grades gesucht, das alle Gleichungen $P(x_i) = y_i$ erfüllt.

Satz 1.2. *Zu beliebigen N Stützpunkten $(x_i, y_i)\,,\ i = 0, \ldots, N$ mit paarweise verschiedenen Stützstellen ($x_i \neq x_j$ für $i \neq j$) existiert genau ein Interpolationspolynom $P_N(x)$, welches die Interpolationsbedingungen erfüllt und höchstens vom Grade N ist.*

Beweis. a) Existenz von $P_N(x)$:

Zum Nachweis der Existenz von $P_N(x)$ werden zunächst die LAGRANGE-Polynome gebildet:

$$L_i(x) = \prod_{\substack{j=0\\ j\neq i}}^{N} \frac{x - x_j}{x_i - x_j} = \frac{(x - x_0)\cdot \ldots \cdot (x - x_{i-1}) \cdot (x - x_{i+1}) \cdot \ldots \cdot (x - x_N)}{(x_i - x_0)\cdot \ldots \cdot (x_i - x_{i-1}) \cdot (x_i - x_{i+1}) \cdot \ldots \cdot (x_i - x_N)} \tag{1.1}$$

Dann gilt:

1) $L_i(x)$ ist ein Polynom vom Grade N.

2) $L_i(x_j) = \delta_{ij} = \begin{cases} 1, \text{ wenn } i = j \\ 0, \text{ wenn } i \neq j \end{cases}$

$\Rightarrow P_N(x) = \sum\limits_{i=0}^{N} y_i \cdot L_i(x)$ ist ein Polynom vom Grade $\leq N$ und erfüllt die Interpolationsbedingungen $P_N(x_j) = x_j, \forall j = 0, 1, \ldots, N$.

b) Eindeutigkeit von $P_N(x)$:

$P_N(x)$ und $Q_N(x)$ seien Polynome vom Grade $\leq N$ mit $P_N(x_i) = Q_N(x_i) = y_i$, $\forall i = 0, 1, \ldots, N$. Sei $D(x) = P_N(x) - Q_N(x)$.

$\Rightarrow$ $D(x)$ ist ein Polynom höchstens N-ten Grades. $D(x)$ besitzt N+1 Nullstellen $x_i\,(i = 0, 1, \ldots, N)$. Widerspruch zum Fundamentalsatz der Algebra[1]
$\Rightarrow D(x) \equiv 0 \Rightarrow Q_N(x) = P_N(x)$ $\square$

Die Darstellung des eindeutig bestimmten Interpolationspolynoms nach Formel (1.1) wird jedoch zumeist nur für Existenz- und Eindeutigkeitssätze verwendet. Zur Berechnung des Polynoms wurden effizientere Methoden entwickelt. Die Wahl des Interpolationsverfahrens richtet sich vor allem danach, ob nur ein Wert interpoliert werden soll oder das Interpolationspolynom gesucht ist. Für den ersten Fall wird in [4] der Algorithmus von AITKEN/NEVILLE vorgeschlagen.
Sei $P^*_{0,1,\ldots,k}$ der Wert des Interpolationspolynoms auf den Stützstellen $x_0, \ldots, x_k$ an der Interpolationsstelle x^*. Dann gilt die folgende Rekursionsformel:

$$P^*_{0,1,\ldots,k} = \frac{(x^* - x_0)\, P^*_{1,\ldots,k} - (x^* - x_k)\, P^*_{0,\ldots,k-1}}{x_k - x_0} \ . \tag{1.2}$$

Zur Berechnung des interpolierten Wertes wird die Formel (1.2) rekursiv angewendet. Dabei gilt:

$$P^*_i = y_i \ \ (i = 0, \ldots, N) \tag{1.3}$$

und

$$f(x^*) = P^*_{0,1,\ldots,N} \ . \tag{1.4}$$

$$\begin{array}{c|c||ccccccccc}
x_i & y_i & & & & & & & & & \\
\hline
x_0 & y_0 & & & & & & & & & \\
 & & \searrow & & & & & & & & \\
x_1 & y_1 & \rightarrow & P^*_{0,1} & & & & & & & \\
 & & \searrow & & \searrow & & & & & & \\
x_2 & y_2 & \rightarrow & P^*_{1,2} & \rightarrow & P^*_{0,1,2} & & & & & \\
 & & \searrow & & \searrow & & \searrow & & & & \\
x_3 & y_3 & \rightarrow & P^*_{2,3} & \rightarrow & P^*_{1,2,3} & \rightarrow & P^*_{0,1,2,3} & & & \\
\vdots & \vdots & & \vdots & & \vdots & & \vdots & \ddots & & \\
x_N & y_N & \rightarrow & P^*_{N-1,N} & \rightarrow & P^*_{N-2,N-1,N} & \rightarrow & P^*_{N-3,N-2,N-1,N} & \cdots & P^*_{0,1,2,3,\ldots,N} &
\end{array}$$

Tabelle 1.1: Interpolationsschema von AITKEN/NEVILLE

[1] Fundamentalsatz der Algebra: Ein Polynom n-ten Grades besitzt über dem Körper der komplexen Zahlen genau N Nullstellen.

Algorithmus 1.1 Interpolationsschema von AITKEN/NEVILLE

```
Eingabe : N+1 Interpolationsstellen x_i, p = [y_0, ..., y_N], Auswertungsstelle x*
Ausgabe : Funktionswert an der Stelle x*
for k = 1 → N do
    for i = N → k do
        p_i = p_i + (x* − x_i) · (p_i − p_{i−1}) / (x_i − x_{i−k});
    end
end
f(x*) = p_N;
```

Benötigt man das Interpolationspolynom und nicht nur den Wert des Polynoms an einer bestimmten Stelle, so empfiehlt sich der Algorithmus von NEWTON. Ähnlich wie beim Algorithmus von AITKEN/NEVILLE wird beim NEWTONschen Algorithmus das Interpolationspolynom rekursiv bestimmt. (vgl. [4] und [16])
Die Koeffizienten des Interpolationspolynoms lassen sich in $O\left(N^2\right)$ bestimmen. Das ermittelte Polynom wiederum lässt sich dann mit $O\left(N\right)$ Operationen an einer Stelle auswerten. Wird das Polynom nicht explizit benötigt, sollte der Algorithmus von AITKEN/NEVILLE verwendet werden, da er weniger anfällig gegenüber Auslöschung[2] ist. Bei beiden Algorithmen können effizient neue Stützstellen eingefügt werden, ohne dass der gesamte Algorithmus erneut ablaufen muss.

1.2.2 Spline-Interpolation

Bei größerer Stützpunktanzahl ist die Interpolation mit einem Polynom über dem gesamten Datenbereich nicht mehr sinnvoll. Deshalb werden dann zumeist Interpolationspolynome auf Teilintervallen gebildet, die dann in geeigneter Art und Weise aneinander gefügt werden müssen. Diese Idee führt zur sogenannten Spline-Interpolation.

Definition 1.3. Ein *Spline* n-ten Grades ist eine Funktion, die stückweise aus Polynomen vom Grade $\leq n$ zusammengesetzt ist, wobei an den Stellen, an denen zwei Polynomstücke aufeinandertreffen, auch Knoten genannt, gewisse Stetigkeits- und Differenzierbarkeitsbedingungen erfüllt sein sollen.

Am weitesten verbreitet ist die kubische Spline-Interpolation. Dabei wird zu N paarweise verschiedenen Stützstellen $x_1 < x_2 < \ldots < x_N$ und zugehörigen Stützwerten $y_1, y_2, \ldots, y_N$ eine Funktion $s\left(x\right)$ gesucht, die folgende Eigenschaften besitzt:

a) $s(x)$ ist auf jedem Teilintervall $[x_i, x_{i+1}]$ $(i = 1, \ldots, N-1)$ ein Polynom 3. Grades und erfüllt die Interpolationsbedingung $s\left(x_i\right) = y_i,\ (i = 1, \ldots, N)$.

[2] Verlust an Genauigkeit bei der Subtraktion fast gleich großer Gleitkommazahlen

b) $s(x)$ ist an jeder inneren Stützstelle x_i, $(i = 2, \ldots, N-1)$ zweimal stetig differenzierbar.

c) $s(x)$ erfüllt die natürlichen Randbedingungen $s''(x_1) = s''(x_N) = 0$.

Mit diesen Forderungen lässt sich $s(x)$ durch Lösen eines tridiagonalen Gleichungssystems eindeutig bestimmen. (vgl. [4] und [16])

Bemerkungen.

- Ist die zu interpolierende Funktion an den Intervallenden sehr stark gekrümmt, so ist die Forderung $s''(x_1) = s''(x_N) = 0$ nicht problemgerecht. Als Ausweg bietet es sich an, jeweils die äußeren drei Punkte durch ein Polynom zu interpolieren, die sogenannten not-a-knot-Randbedingungen.

- Aufgrund ihrer Konstruktion neigen Splines im Gegensatz zu Interpolationspolynomen nicht zum Überschwingen.

1.3 Approximationsverfahren für eindimensionale Daten

Bislang wurde zu gegebenen Stützpunkten (x_i, y_i) eine Funktion gesucht, deren Funktionswerte an den Stützstellen exakt mit den Stützwerten übereinstimmen. Diese Forderung der Interpolation wird nun fallen gelassen. Stattdessen wird verlangt, dass die Näherungsfunktion die gegebenen Daten möglichst gut approximiert, wobei noch geklärt werden muss, was unter „möglichst gut“ zu verstehen ist.
Bei der Approximation unterscheidet man grundsätzlich zwei Aufgabenstellungen:

1. Eine gegebene Funktion f soll durch eine Funktion Φ ersetzt werden, die sich zweckmäßiger behandeln lässt, d.h. die sich z. B. besser integrieren oder differenzieren lässt. Hierbei spricht man von *kontinuierlicher Approximation.*

2. Eine empirische Funktion, von der nur endlich viele Wertepaare (x_i, y_i) bekannt sind, soll durch eine Funktion f ersetzt werden. Man spricht von *diskreter Approximation.*

Die zweite Aufgabenstellung bildet den relevanten Fall für die weiteren Untersuchungen. Im Folgenden wird die *diskrete Approximation im quadratischen Mittel* untersucht. Zu gegebenen Stützpunkten (x_i, y_i), $i = 1, \ldots, N$ wird für die Approximationsfunktion f der folgende Ansatz gewählt:

$$f(x) = \sum_{j=1}^{n} c_j \varphi_j(x), \;\; n \leq N \; . \tag{1.5}$$

Dabei sind φ_j vorgegebene Basisfunktionen und c_j noch zu bestimmende Koeffizienten. Zur Bestimmung der Größen c_j wird der quadratische Fehler an den Stützstellen minimiert. Zunächst werden sogenannte Approximationsgewichte $w_i = {v_i}^2$ $(v_i > 0,\ i = 1, \ldots, N)$ eingeführt. Die Aufgabe zur Bestimmung der c_j lautet dann:

$$\Delta_{SQ}\left(c_1, \ldots, c_n\right) = \sum_{i=1}^{N} w_i \left(f\left(x_i\right) - y_i\right)^2 = \sum_{i=1}^{N} w_i \left(\sum_{j=1}^{n} c_j \varphi_j(x) - y_i\right)^2 \rightarrow \underset{c_1,\ldots,c_n}{\mathrm{Min}} \quad . \tag{1.6}$$

Mit $\mathbf{\Phi} = \begin{pmatrix} v_1\varphi_1\left(x_1\right) & v_1\varphi_2\left(x_1\right) & \cdots & v_1\varphi_n\left(x_1\right) \\ v_2\varphi_1\left(x_2\right) & v_2\varphi_2\left(x_2\right) & \cdots & v_2\varphi_n\left(x_2\right) \\ \vdots & \vdots & \ddots & \vdots \\ v_N\varphi_1\left(x_N\right) & v_N\varphi_2\left(x_N\right) & \cdots & v_N\varphi_n\left(x_N\right) \end{pmatrix}, \mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}$ und $\mathbf{b} = \begin{pmatrix} v_1 y_1 \\ v_2 y_2 \\ \vdots \\ v_N y_N \end{pmatrix}$

erhält man aus (1.6) das Minimierungsproblem:

$$\Delta_{SQ}\left(\mathbf{c}\right) = {\left\|\mathbf{\Phi c} - \mathbf{b}\right\|_2}^2 \rightarrow \underset{\mathbf{c}\in\mathbb{R}^n}{\mathrm{Min}} \quad . \tag{1.7}$$

Die notwendigen Bedingungen für ein Extremum von $\Phi\left(\mathbf{c}\right)$

$$\frac{\partial \Delta_{SQ}}{\partial c_k} = 2\sum_{i=1}^{N} w_i \left(\sum_{j=1}^{n} c_j \varphi_j\left(x_i\right) - y_i\right) \varphi_k\left(x_i\right) \stackrel{!}{=} 0 \tag{1.8}$$

liefern n lineare Gleichungen zur Bestimmung der n Koeffizienten c_j. Zusammengefasst in Matrixform ergibt sich

$$\frac{\partial \Delta_{SQ}}{\partial \mathbf{c}} = 2 \cdot \left(\mathbf{\Phi}^T\mathbf{\Phi c} - \mathbf{\Phi}^T\mathbf{b}\right) \stackrel{!}{=} 0 \tag{1.9}$$

und damit das *Gaußsche Normalgleichungssystem*:

$$\mathbf{\Phi}^T\mathbf{\Phi c} = \mathbf{\Phi}^T\mathbf{b} \ . \tag{1.10}$$

Die hinreichende Bedingung für ein Minimum von $\Delta_{SQ}\left(\mathbf{c}\right)$ ist erfüllt, wenn die Funktionalmatrix $\mathbf{H} = \left\{\frac{\partial^2\Delta_{SQ}}{\partial c_k \partial c_l}\right\}_{k,l=1}^{n}$ positiv definit ist. Dies ist genau dann der Fall, wenn der Rang der Matrix $\mathbf{\Phi}$ gleich n ist.

Definition 1.4. Das Funktionensystem $\varphi_1\left(x\right), \ldots, \varphi_n\left(x\right)$ heißt *Tschebyschew-System*, wenn jedes verallgemeinerte Polynom $a_1\varphi_1\left(x\right) + a_2\varphi_2\left(x\right) + \ldots + a_n\varphi_n\left(x\right) \not\equiv 0$ weniger als n Nullstellen besitzt.

Satz 1.5. *Zu beliebigen Approximationspunkten* (x_i, y_i) , $i = 1, \ldots, N$ *mit paarweise verschiedenen Stützstellen existiert für jedes Tschebyschew-System* $\varphi_1, \ldots, \varphi_n$ *mit* $n \leq N$ *genau eine Approximationsfunktion, welche die Approximationsbedingung erfüllt.*

Beweis. $\varphi_1, \ldots, \varphi_n$ ist ein Tschebyschew-System $\Rightarrow$ $\mathbf{\Phi c} = \mathbf{0}$ besitzt nur die triviale Lösung $\mathbf{c} = \mathbf{0}$ $\Rightarrow$ rang($\mathbf{\Phi}$)$= n$. □

Bemerkungen.

- Um numerische Instabilität zu vermeiden, sollte an Stelle der direkten Lösung des Normalgleichungssytems die Lösung des Minimumproblems mittels des Verfahrens von HOUSEHOLDER bestimmt werden.
- Wenn keine weiteren Informationen über die Daten bekannt sind, so werden zumeist die Approximationsgewichte $w_i = 1, \ \forall i$ gewählt. Eine andere Wahl der Gewichte ist sinnvoll, wenn die einzelnen Datenpunkte unterschiedlichen Genauigkeiten unterliegen. Dann können weniger genauen Funktionswerten kleinere Gewichte zugeordnet werden.

Das Verfahren von HOUSEHOLDER

Das Verfahren von HOUSEHOLDER kann verwendet werden, um die Normallösung eines überbestimmten linearen Gleichungssystems zu bestimmen.

Definition 1.6. Der Vektor $\mathbf{x}^* \in \mathbb{R}^n$ heißt Normallösung des überbestimmten Gleichungsystems $\mathbf{Ax} = \mathbf{b}$, wenn

$$\|\mathbf{Ax}^* - \mathbf{b}\|_2{}^2 = \underset{\mathbf{x}\in\mathbb{R}^n}{\mathrm{Min}} \|\mathbf{Ax} - \mathbf{b}\|_2{}^2 \ . \tag{1.11}$$

Bemerkung. Das Minimierungsproblem (1.7) entspricht der Bestimmung der Normallösung des überbestimmten Gleichungsystems.

Das Verfahren von HOUSEHOLDER basiert auf dem folgenden Satz (vgl. [4]):

Satz 1.7. *Sei* $\mathbf{A}$ *eine* $m \times n$ *Matrix* $m \geq n$ *und* $\mathbf{rang}\,(\mathbf{A}) = n$. *Dann existiert eine orthogonale* $m \times m$ *Matrix* $\mathbf{Q}$, *sodass gilt:*

$$\mathbf{A} = \mathbf{Q} \cdot \mathbf{R} \ , \tag{1.12}$$

wobei $\mathbf{R}$ *eine obere* $m \times n$ *Dreiecksmatrix ist.*

Definition 1.8. Eine quadratische Matrix $\mathbf{Q}$ heißt orthogonal, wenn

$$\mathbf{Q}^T\mathbf{Q} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}\ . \tag{1.13}$$

Bemerkungen. Für eine orthogonale Matrix gilt:

- $\mathbf{Q}^T = \mathbf{Q}^{-1}$
- $\|\mathbf{Qx}\| = \|\mathbf{x}\|$
- $\langle \mathbf{Qx}, \mathbf{Qy} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle,\ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

Damit lassen sich die Ausdrücke in Gleichung (1.11) umformen:

$$\begin{aligned} {\|\mathbf{Ax} - \mathbf{b}\|_2}^2 &= {\|\mathbf{Q}^T\left(\mathbf{Ax} - \mathbf{b}\right)\|_2}^2 \\ &= {\|\mathbf{Q}^T\mathbf{Ax} - \mathbf{Q}^T\mathbf{b}\|_2}^2 \\ &= {\|\mathbf{Rx} - \mathbf{Q}^T\mathbf{b}\|_2}^2 \end{aligned} \tag{1.14}$$

und erhält damit das lineare Gleichungssystem:

$$\mathbf{Rx} = \mathbf{Q}^T\mathbf{b}\ . \tag{1.15}$$

Da $\mathbf{R}$ eine obere $m \times n$ Dreiecksmatrix ist, lässt sich das Gleichungsystem noch umformen. Sei

$$\mathbf{R} = \begin{pmatrix} \hat{\mathbf{R}} \\ \mathbf{0} \end{pmatrix} \quad \text{und } \mathbf{Q}^T\mathbf{b} = \begin{pmatrix} \mathbf{r}^{(1)} \\ \mathbf{r}^{(2)} \end{pmatrix},$$

wobei $\hat{\mathbf{R}}$ eine obere $n \times n$ Dreiecksmatrix, $\mathbf{r}^{(1)}$ ein Vektor der Länge n und $\mathbf{r}^{(2)}$ ein Vektor der Länge $m - n$ ist.
Dann ergibt sich die Normallösung $\mathbf{x}^*$ des überbestimmten Gleichungssystems als Lösung von:

$$\hat{\mathbf{R}}\mathbf{x} = \mathbf{r}^{(1)}\ . \tag{1.16}$$

Nun gilt es eine geignete orthogonale Matrix $\mathbf{Q}$ zu finden. Dafür wird die sogenannte HOUSEHOLDER-Transformation verwendet. Dabei wird die Matrix $\mathbf{A}$ durch sukzessive Linksmultiplikation mit HOUSEHOLDER-Matrizen in eine obere Dreiecksmatrix $\mathbf{R}$ überführt. Die Konstruktionsvorschrift der HOUSEHOLDER-Matrizen findet man in [4].

1.4 Mehrdimensionale Approximation

Bislang wurden Interpolations- und Approximationsverfahren für eindimensionale Daten untersucht. In der Praxis liefern technische Anwendung jedoch zumeist mehrdimensionale Daten, d.h. die gesuchten Funktionen $f : D \subseteq \mathbb{R}^m \to \mathbb{R}^n$ sind von mehreren Variablen abhängig. Da die Datenpunkte teilweise Messungenauigkeiten unterliegen, ist es oftmals viel wichtiger den Trend der Funktion widerzuspiegeln, als die Interpolationseigenschaft zu erfüllen. Da die Interpolation auch nur einen Spezialfall der Approximation darstellt, wird im Folgenden zumeist nur noch von mehrdimensionaler Approximation gesprochen, selbst wenn ein Verfahren Funktionen liefert, welche die Interpolationsbedingungen erfüllen.

Bei der mehrdimensionalen Approximation ist die Verteilung, Lage und Anzahl der Datenpunkte sehr entscheidend. Liegen die Daten auf einem Gitter, welches möglicherweise auch noch äquidistant ist, so können spezielle Lösungsverfahren verwendet werden, die genau diese Eigenschaften berücksichtigen und nutzen. Bei vielen praktischen Anwendungen kann jedoch davon ausgegangen werden, dass die Daten unregelmäßig im m-dimensionalen Raum verteilt sind.

2 Approximationsverfahren für mehrdimensionale Daten

In diesem Kapitel werden Approximationsverfahren für Daten aus einem prinzipiell beliebig dimensionalen Euklidischen Raum $\mathbb{R}^m$ vorgestellt. Zunächst wird die genaue Approximationsaufgabe formuliert, sowie die Bezeichnungen festgelegt. Anschließend wird die Approximation eindimensionaler Daten auf den mehrdimensionalen Fall verallgemeinert. Zudem werden Verfahren mit speziellen Ansätzen und Ideen untersucht.

2.1 Formulierung der Approximationsaufgabe

Im Folgenden wird zu gegebenen Stützstellen $\mathbf{x}^{(i)} \in \mathbb{R}^m$ und dazugehörigen Stützwerten $\mathbf{y}^{(i)} \in \mathbb{R}^n$ eine Funktion $f : D \subseteq \mathbb{R}^m \to \mathbb{R}^n$ gesucht, die den Zusammenhang der gegebenen Daten möglichst gut approximiert. Um das Problem nicht unnötig zu verkomplizieren, wird im Folgenden nur der Fall $n = 1$ betrachtet. Diese Annahme birgt keine Einschränkungen, da für den Fall $n \geq 2$ n separate Approximationsmodelle aufgestellt werden können. Da in der Approximationsbibliothek der IAV die Daten in alle Dimensionen auf das Intervall $[-1, 1]$ transformiert werden, wird dementsprechend eine Funktion $f : [-1, 1]^m \to \mathbb{R}$ gesucht.
Ist eine Schätzung des Funktionswertes an einer Stelle $\mathbf{x} \in \mathbb{R}^m$ gesucht, so wird diese mit $\mathbf{x}^*$ bezeichnet und Auswertungsstelle genannt.

2.2 Mehrdimensionale Approximation im quadratischen Mittel

Zur Entwicklung eines Verfahrens zur mehrdimensionalen Approximation wird zunächst untersucht, ob sich das Vorgehen im eindimensionalen Fall auf mehrdimensionale Daten übertragen lässt. Es gilt zunächst die Approximationsgewichte sowie Basisfunktionen zu wählen. Da im Allgemeinen keine Informationen über die Genauigkeit der Messdaten vorliegen, werden die Approximationsgewichte wie in 1.3 diskutiert identisch eins gesetzt. Bei der Wahl der Basisfunktionen ist darauf zu achten, dass die Gaußschen

Normalgleichungssysteme mit wachsendem N sehr schnell schlecht konditioniert werden.

Definition 2.1. Ein Problem nennt man *schlecht konditioniert*, wenn kleine Änderungen in den Eingangsdaten (unvermeidbare Messfehler) zu großen Änderungen in den Ausgangsdaten führen.

Um diesen Effekt zu vermeiden, sollten anstelle der gewöhnlichen Basisfunktionen x^j orthogonale Polynome verwendet werden.

2.2.1 Orthogonale Polynome

Der Begriff der Orthogonalität ist aus der linearen Algebra bekannt. So werden zwei Vektoren orthogonal zueinander genannt, wenn ihr Skalarprodukt null ist. Auch die Menge der integrierbaren Funktionen lässt sich als Vektorraum auffassen. Mit Hilfe von

$$\langle f, g\rangle := \int_a^b f(x)\, g(x)\, w(x)\, dx \tag{2.1}$$

kann man für ein beliebiges Intervall $[a, b]$ ein Skalarprodukt definieren. Dabei ist $w(x)$ eine auf $[a, b]$ nichtnegative Gewichtsfunktion. Im Folgenden wird $w(x) \equiv 1$ gesetzt.
Es lässt sich zeigen, dass $\langle f, g\rangle$ den Eigenschaften eines Skalarproduktes genügt und insbesondere mit

$$\|f\| := \sqrt{\langle f, f\rangle} \tag{2.2}$$

die Eigenschaften der Norm besitzt.
Orthogonale Polynome lassen sich analog zu [9] wie folgt definieren:

Definition 2.2. Ist $\langle f, g\rangle$ ein Skalarprodukt auf dem Intervall $[a, b]$, so nennt man eine Familie von Funktionen $\varphi_k : [a, b] \to \mathbb{R}\,(k \in \mathbb{N}_0)$ ein *Orthogonalsystem* (OS), wenn $\langle \varphi_k, \varphi_l\rangle = 0$ für $k \neq l$ und $\|\varphi_k\|^2 = \langle \varphi_k, \varphi_k\rangle \neq 0$ für $k \in \mathbb{N}_0$ gilt. Gilt ferner $\|\varphi_k\|^2 = 1$ für $k \in \mathbb{N}_0$, so wird das System als *Orthonormalsystem* (ONS) bezeichnet. Sind die Funktionen φ_k Polynome vom Grade k, so nennt man die Funktionen *orthogonale Polynome.*

Satz 2.3. *Zu jedem Skalarprodukt $\langle \cdot, \cdot\rangle$ nach* (2.1) *gibt es eindeutig bestimmte Orthogonalpolynome $\varphi_k \in P_k$ mit dem führenden Koeffizienten Eins. Diese Polynome erfüllen* *die Drei-Term-Rekursion:*

$$\varphi_k(x) = (x + \alpha_k)\,\varphi_{k-1}(x) + \beta_k \varphi_{k-2}(x), \quad k = 1, 2, \ldots \tag{2.3}$$

mit $\varphi_{-1} := 0,\ \varphi_0 := 1$ *und*

$$\alpha_k = -\frac{\langle x\varphi_{k-1}, \varphi_{k-1}\rangle}{\langle \varphi_{k-1}, \varphi_{k-1}\rangle},\ \ \beta_k = -\frac{\langle \varphi_{k-1}, \varphi_{k-1}\rangle}{\langle \varphi_{k-2}, \varphi_{k-2}\rangle}\ . \tag{2.4}$$

Durch Normierung der erhaltenen Polynome entstehen die eindeutig bestimmten Orthonormalpolynome.

Beweis. Unmittelbar sieht man, dass $\varphi_0 \equiv 1 \in P_0$, der Rest der Behauptung wird induktiv gezeigt.
Seien $\varphi_0, \ldots, \varphi_{k-1}$ paarweise orthogonale Polynome mit $\varphi_l \in P_l$ von Grade l und führendem Koeffizienten Eins. Ist $\varphi_k \in P_k$ auf gleiche Weise normiert, so folgt

$$\varphi_k(x) - x\varphi_{k-1}(x) \ \text{ist vom Grade } \leq k-1.$$

Da $\varphi_0, \ldots, \varphi_{k-1}$ eine Orthogonalbasis von P_{k-1} bzgl. $\langle \cdot, \cdot \rangle$ bilden, gilt

$$\varphi_k(x) - x\varphi_{k-1}(x) = \sum_{l=0}^{k-1} c_l \varphi_l \ \text{mit}\ c_l = \frac{\langle \varphi_k - x\varphi_{k-1}, \varphi_l\rangle}{\langle \varphi_l, \varphi_l\rangle} = \frac{\langle \varphi_k, \varphi_l\rangle}{\langle \varphi_l, \varphi_l\rangle} - \frac{\langle x\varphi_{k-1}, \varphi_l\rangle}{\langle \varphi_l, \varphi_l\rangle}$$

Da zudem φ_k orthogonal zu $\varphi_0, \cdots, \varphi_{k-1}$ sein soll, folgt

$$c_l = -\frac{\langle x\varphi_{k-1}, \varphi_l\rangle}{\langle \varphi_l, \varphi_l\rangle} = -\frac{\langle \varphi_{k-1}, x\varphi_l\rangle}{\langle \varphi_l, \varphi_l\rangle}$$

und weil das Produkt $x\varphi_l$ als Linearkombination von $\varphi_0, \ldots, \varphi_{l+1} \in P_{l+1}$ geschrieben werden kann, gilt für $l+1 < k-1$ $c_l = 0$, also

$$c_0 = \ldots = c_{k-3} = 0. \tag{2.5}$$

Somit erhält man:

$$c_{k-1} = -\frac{\langle x\varphi_{k-1}, \varphi_{k-1}\rangle}{\langle \varphi_{k-1}, \varphi_{k-1}\rangle}, \tag{2.6}$$

sowie unter Verwendung von $x\varphi_{k-2} = \varphi_{k-1} + d_{k-2}\varphi_{k-2} + \ldots + d_0\varphi_0$

$$c_{k-2} = -\frac{\langle \varphi_{k-1}, x\varphi_{k-2}\rangle}{\langle \varphi_{k-2}, \varphi_{k-2}\rangle} = -\frac{\langle \varphi_{k-1}, \varphi_{k-1}\rangle}{\langle \varphi_{k-2}, \varphi_{k-2}\rangle} \tag{2.7}$$

und damit die Drei-Term-Rekursion. □

Da in der Approximationsbibliothek der IAV die gegebenen Daten in allen Dimensionen auf das Intervall $[-1, 1]$ normiert werden, gilt es auf diesem Intervall die eindeutig bestimmte Familie von orthonormalen Polynomen mit der Gewichtsfunktion $w(x) \equiv 1$ zu bestimmen. Als Ergebnis erhält man die normierten LEGENDRE-Polynome:

$$\begin{aligned}
\varphi_0(x) &= 1 \\
\varphi_1(x) &= \sqrt{\tfrac{3}{2}}x \\
\varphi_2(x) &= \sqrt{\tfrac{5}{8}}\left(3x^2 - 1\right) \\
\varphi_3(x) &= \sqrt{\tfrac{7}{8}}\left(5x^3 - 3x\right) \\
\varphi_4(x) &= \sqrt{\tfrac{9}{128}}\left(35x^4 - 30x^2 + 3\right) \\
\varphi_5(x) &= \sqrt{\tfrac{11}{128}}\left(63x^5 - 70x^3 + 15x\right) \\
\varphi_6(x) &= \sqrt{\tfrac{13}{512}}\left(231x^6 - 315x^4 + 105x^2 - 5\right)
\end{aligned}$$

2.2.2 Tensorprodukt

Im Abschnitt 2.2.1 wurden eindimensionale orthogonale Basisfunktionen konstruiert. Diese sollen im Folgenden dazu verwendet werden, orthogonale Basisfunktionen auf $[-1, 1]$ zu erzeugen. Dazu wird in [1] vorgeschlagen, Tensorprodukte der eindimensionalen Basisfunktionen zu bilden. Mit Hilfe eines m-dimensionalen Vektors $\mathbf{z} = (z_1, \ldots, z_m)$ von nichtnegativen ganzen Zahlen wird das Tensorprodukt wie folgt berechnet:

$$\Phi_{z_1,\ldots,z_m}(\mathbf{x}) = \prod_{j=1}^{m} \varphi_{z_j}(x_j) \quad . \tag{2.8}$$

Die Anzahl der Funktionen Φ kann durch zusätzliche Eigenschaften des Vektors $\mathbf{z}$ variiert werden. Dazu werden der Maximalgrad $\mathfrak{G}$, Maximalrang $\mathfrak{R}$ und die Maximalordnung $\mathfrak{O}$ eingeführt. Für $\mathbf{z}$ muss dann gelten:

$$\sum_{j=1}^{m} z_j \leq \mathfrak{G} \ , \tag{2.9}$$

$$\sum_{j=1}^{m} 1_{z_j \neq 0} \leq \mathfrak{R} \ , \tag{2.10}$$

$$\max_{1 \leq j \leq m} z_j \leq \mathfrak{O} \ . \tag{2.11}$$

Nachdem diese Werte manuell eingestellt werden, können sowohl die Anzahl der verschiedenen Vektoren $\mathbf{z}$ als auch die Matrix $\mathbf{Z}$, die alle möglichen Vektoren $\mathbf{z}$ enthält, rekursiv berechnet werden.

Algorithmus 2.1 Berechnung der Tensorprodukte

```
Eingabe : 𝔊, ℜ, 𝔒
Initialisierung : z = 0, Start = 0, Rank = 0, Row = -1;
Z = Tensorprodukt (z, Start, Rank, Z, 𝔊, ℜ, 𝔒) {
    Row++;
    Z [Row, :] = z;
    if Rank < ℜ then
        Sum = z.VectorSum();
        if Sum ≥ 𝔊 then
            return Z;
        end
        for i = Start → m do
            if z [i] = 0 then
                for j = 1 → 𝔒 do
                    if Sum + j ≤ 𝔊 then
                        z [i] = j;
                        Z = Tensorprodukt (z, i + 1, Rank + 1, Z, 𝔊, ℜ, 𝔒);
                    end
                    else
                        break;
                    end
                end
                z [i] = 0;
            end
        end
    end
    return Z;
}
```

2.2.3 Regression

Ähnlich wie in [1] wird die mehrdimensionale Approximation im quadratischen Mittel im Folgenden als Regression bezeichnet. In Analogie zum eindimensionalen Fall in 1.3 wird der folgende Ansatz für die Approximationsfunktion gewählt:

$$f(\mathbf{x}) = \sum_{j=1}^{n} c_j \phi^{(j)}(\mathbf{x}) \quad . \tag{2.12}$$

Dabei sind $\phi^j(\mathbf{x})$ mehrdimensionale Basisfunktionen, für die gelten soll:

$$\phi^{(1)}(\mathbf{x}) = 1, \quad \forall \mathbf{x} \in [-1,1]^m \ , \tag{2.13}$$

$$\int \phi^{(j)}(\mathbf{x})\, d\mathbf{x} = 0, \quad j > 1 \ , \tag{2.14}$$

$$\int \phi^{(j)}(\mathbf{x})\, \phi^{(k)}(\mathbf{x})\, d\mathbf{x} = \delta_{jk} \ , \tag{2.15}$$

wobei alle Integrale über $\mathbf{x} \in [-1,1]^m$ zu verstehen sind und c_j noch zu bestimmende Koeffizienten.
Wählt man die Basisfunktionen $\phi^{(j)}(\mathbf{x})$ als Tensorprodukte der eindimensionalen normierten LEGENDRE-Polynome, so sind die gestellten Bedingungen erfüllt. Wie leicht zu sehen ist, gilt dann $\phi^{(1)}(\mathbf{x}) = \Phi_{0,\ldots,0}(\mathbf{x}) \equiv 1$.
Analog zu 1.3 erhält man wieder das Gaußsche Normalgleichungssystem.

Mit $\mathbf{\Phi} = \begin{pmatrix} \phi^{(1)}\left(\mathbf{x}^{(1)}\right) & \phi^{(2)}\left(\mathbf{x}^{(1)}\right) & \cdots & \phi^{(n)}\left(\mathbf{x}^{(1)}\right) \\ \phi^{(1)}\left(\mathbf{x}^{(2)}\right) & \phi^{(2)}\left(\mathbf{x}^{(2)}\right) & \cdots & \phi^{(n)}\left(\mathbf{x}^{(2)}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \phi^{(1)}\left(\mathbf{x}^{(N)}\right) & \phi^{(2)}\left(\mathbf{x}^{(N)}\right) & \cdots & \phi^{(n)}\left(\mathbf{x}^{(N)}\right) \end{pmatrix}$, $\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}$ und $\mathbf{b} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{pmatrix}$

ergibt sich:

$$\mathbf{\Phi}^T \mathbf{\Phi} \mathbf{c} = \mathbf{\Phi}^T \mathbf{b} \ . \tag{2.16}$$

Das Gleichungssystem kann mittels des Cholesky-Verfahrens gelöst werden. Es empfiehlt sich jedoch ähnlich wie in 1.3 die Berechnung der Normallösung des überbestimmten Gleichungssystems $\mathbf{\Phi c} = \mathbf{b}$ mit Hilfe des Verfahrens von HOUSEHOLDER.

2.2.4 Quasi-Regression

Bei genauerer Untersuchung der Matrix $\mathbf{\Phi}^T\mathbf{\Phi}$ lässt sich eine Idee entwickeln, wie man die Approximation beschleunigen kann. Es gilt:

$$\mathbf{\Phi}^T\mathbf{\Phi} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \phi^{(2)}\left(\mathbf{x}^{(1)}\right) & \phi^{(2)}\left(\mathbf{x}^{(2)}\right) & \cdots & \phi^{(2)}\left(\mathbf{x}^{(N)}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \phi^{(n)}\left(\mathbf{x}^{(1)}\right) & \phi^{(n)}\left(\mathbf{x}^{(2)}\right) & \cdots & \phi^{(n)}\left(\mathbf{x}^{(N)}\right) \end{pmatrix} \cdot \begin{pmatrix} 1 & \phi^{(2)}\left(\mathbf{x}^{(1)}\right) & \cdots & \phi^{(n)}\left(\mathbf{x}^{(1)}\right) \\ 1 & \phi^{(2)}\left(\mathbf{x}^{(2)}\right) & \cdots & \phi^{(n)}\left(\mathbf{x}^{(2)}\right) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \phi^{(2)}\left(\mathbf{x}^{(N)}\right) & \cdots & \phi^{(n)}\left(\mathbf{x}^{(N)}\right) \end{pmatrix}$$

$$= \begin{pmatrix} N & \sum\limits_{i=1}^{N}\phi^{(2)}\left(\mathbf{x}^{(i)}\right) & \cdots & \sum\limits_{i=1}^{N}\phi^{(n)}\left(\mathbf{x}^{(i)}\right) \\ \sum\limits_{i=1}^{N}\phi^{(2)}\left(\mathbf{x}^{(i)}\right) & \sum\limits_{i=1}^{N}\left(\phi^{(2)}\left(\mathbf{x}^{(i)}\right)\right)^2 & \cdots & \sum\limits_{i=1}^{N}\phi^{(2)}\left(\mathbf{x}^{(i)}\right)\cdot\phi^{(n)}\left(\mathbf{x}^{(i)}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \sum\limits_{i=1}^{N}\phi^{(n)}\left(\mathbf{x}^{(i)}\right) & \sum\limits_{i=1}^{N}\phi^{(n)}\left(\mathbf{x}^{(i)}\right)\cdot\phi^{(2)}\left(\mathbf{x}^{(i)}\right) & \cdots & \sum\limits_{i=1}^{N}\left(\phi^{(n)}\left(\mathbf{x}^{(i)}\right)\right)^2 \end{pmatrix}. \tag{2.17}$$

Aufgrund der Orthonormalität der Basisfunktionen $\phi^{(j)}(\mathbf{x})$, nähern sich die Summen auf der Hauptdiagonale immer mehr dem Wert $\frac{N}{2}$ an, die Nichthauptdiagonalelemente der Null. Die Summen können als einfachste Integralschätzmethode betrachtet werden. Dies soll das folgende Beispiel veranschaulichen:

Beispiel 2.4. Anhand des normierten LEGENDRE-Polynoms zweiten Grades soll die Abschätzung der Summen veranschaulicht werden:

$$\varphi_2(x) = \sqrt{\tfrac{5}{8}}\left(3x^2 - 1\right)$$

Untersucht wird $\sum\limits_{i=1}^{N}{\varphi_2}^2$, wobei aus der Orthonormalität bekannt ist, dass $\int\limits_{i=1}^{N}{\varphi_2}^2 = 1$. Die folgende Abbildung soll die Idee der Abschätzung veranschaulichen.

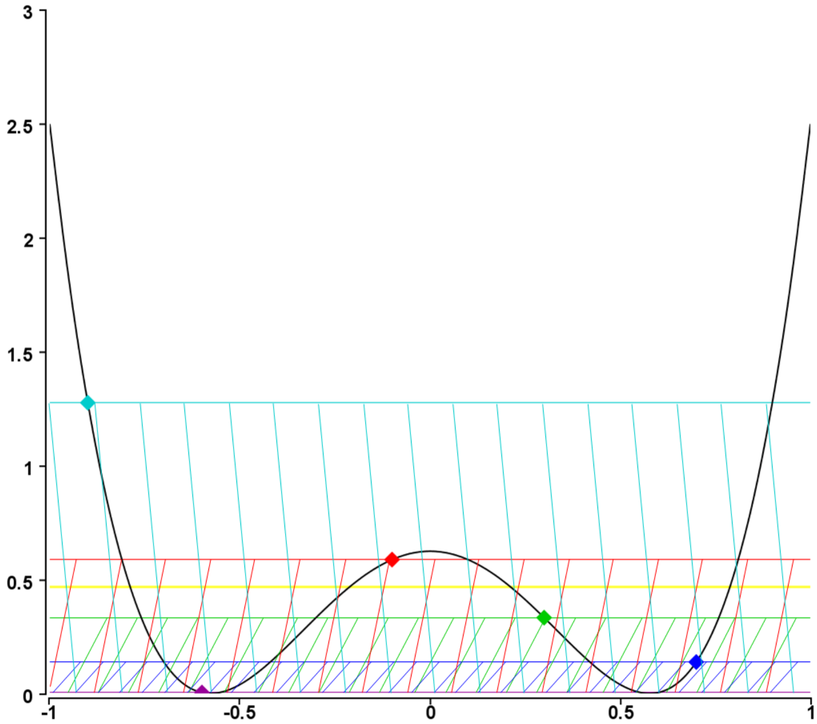

Abbildung 2.1: Abschätzung der Summen

Die in der Abbildung gelb dargestellte Linie zeigt, wie nah der Mittelwert der Funktionswerte an 0.5 liegt. Wählt man 10 Punkte zufällig im Intervall $[-1, 1]$, so ergibt sich bei 10000 Durchläufen im Mittel ein Wert für die untersuchte Summe, der nicht mehr als 0.3 vom geschätzten Wert 5 abweicht.

Diese Methode der Schätzung lässt sich analog auf die mehrdimensionalen Basisfunktionen übertragen. Betrachtet man die Auswertung der Summen auf der Hauptdiagonale von $\mathbf{\Phi}^T\mathbf{\Phi}$ noch etwas genauer, so erkennt man, das sich diese für große N allgemein dem Wert $\frac{N}{l}$ annähern, wobei l die Länge des Intervalls ist. Normiert man die Daten also nicht auf das Interval $[-1, 1]$, sondern z. B. auf $\left[-\frac{1}{2}, \frac{1}{2}\right]$, so lässt sich folgende Abschätzung für große N treffen:

$$\mathbf{\Phi}^T\mathbf{\Phi} \approx N \cdot \mathbf{I} \tag{2.18}$$

und damit:

$$\left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1} \approx \frac{1}{N} \cdot \mathbf{I} \,. \tag{2.19}$$

Diese Idee wird in [1] verfolgt und Quasi-Regression genannt.
Für die Orthonormalpolynome auf $\left[-\frac{1}{2}, \frac{1}{2}\right]$ ergibt sich:

$$
\begin{aligned}
\varphi_0(x) &= 1 \\
\varphi_1(x) &= \sqrt{12}x \\
\varphi_2(x) &= \sqrt{180}\left(x^2 - \tfrac{1}{12}\right) \\
\varphi_3(x) &= \sqrt{2800}\left(x^3 - \tfrac{3}{20}x\right) \\
\varphi_4(x) &= 210\left(x^4 - \tfrac{3}{14}x^2 + \tfrac{3}{560}\right) \\
\varphi_5(x) &= 252\sqrt{11}\left(x^5 - \tfrac{5}{18}x^3 + \tfrac{5}{336}x\right) \\
\varphi_6(x) &= 924\sqrt{13}\left(x^6 - \tfrac{15}{44}x^4 + \tfrac{5}{176}x^2 - \tfrac{5}{14784}\right)
\end{aligned}
$$

Während bei der Regression das Gaußsche Normalgleichungssystem zu lösen ist und der Vektor $\mathbf{c}$ geschätzt wird durch:

$$\mathbf{c} = \left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^T\mathbf{b} \; , \tag{2.20}$$

lässt sich durch die Abschätzung (2.19) der Rechenaufwand deutlich verringern. Es gilt:

$$\tilde{\mathbf{c}} = \frac{1}{N}\mathbf{\Phi}^T\mathbf{b} \; . \tag{2.21}$$

2.2.5 Moving Least Squares

Die bisher analysierten Verfahren Regression und Quasi-Regression hatten das Ziel den globalen quadratischen Fehler der Approximation zu minimieren. Analog dazu gibt es in [6] und [8] auch Ideen, die Approximation im quadratischen Mittel zu lokalisieren. Im Gegensatz zur Regression, die den Ansatz

$$f(\mathbf{x}) = \sum_{j=1}^{n} c_j \phi^{(j)}(\mathbf{x}) \tag{2.22}$$

nutzte, werden beim Verfahren Moving Least Squares die skalaren Koeffizienten nicht mehr global bestimmt, sondern lokal. Die Approximationsfunktion wird definiert als:

$$f(\mathbf{x}) = \sum_{j=1}^{n} c_j(\mathbf{x})\, \phi^{(j)}(\mathbf{x}) \; . \tag{2.23}$$

Ähnlich wie bei der Regression könnten die Koeffizienten $c_j(\mathbf{x})$ mit der Kleinste-Quadrate-Approximation bestimmt werden. Um jedoch ein lokales Approximationsmodell zu gewährleisten wird eine Gewichtsfunktion eingeführt. Liegt eine Auswertungsstelle $\mathbf{x}^*$ sehr nah bei einer Stützstelle $\mathbf{x}^{(i)}$, so sollte der Wert an dieser Stelle stärker gewichtet werden. Dazu wird zunächst der Abstand $r_i = \|\mathbf{x}^* - \mathbf{x}^{(i)}\|$ berechnet und eine Funktion gesucht, die ihr Maximum bei $r = 0$ besitzt und dessen Funktionswerte mit wachsendem Abstand recht schnell kleiner werden. In [18] wird die Gaußfunktion als Gewichtsfunktion vorgeschlagen. Es gilt dann:

$$w_{exp}\left(\mathbf{x}^{(i)}\right) = exp\left(-\frac{\|\mathbf{x}^* - \mathbf{x}^{(i)}\|^2}{\lambda^2}\right) . \tag{2.24}$$

Dabei lässt sich mit Hilfe des Parameters λ^2 der Einflussbereich einer Stützstelle variieren.
Mit Hilfe der Gewichtsfunktion lässt sich nun das Approximationsschema des Verfahrens Moving Least Squares aufstellen:

$$f\left(\mathbf{x}^*\right) = \phi^T\left(\mathbf{x}^*\right)\left(\mathbf{\Phi}^T\mathbf{W}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^T\mathbf{W}\mathbf{b} , \tag{2.25}$$

wobei

$$\begin{aligned}\phi\left(\mathbf{x}^*\right) &= \left(\phi^{(1)}\left(\mathbf{x}^*\right), \ldots, \phi^{(n)}\left(\mathbf{x}^*\right)\right) ,\\ \mathbf{W} &= diag\left(w\left(\mathbf{x}^{(1)}\right), \ldots, w\left(\mathbf{x}^{(N)}\right)\right) ,\\ \mathbf{b} &= \left(y^{(1)}, \ldots, y^{(N)}\right)^T .\end{aligned}$$

Es gilt also:

$$\mathbf{c}\left(\mathbf{x}^*\right) = \left(\mathbf{\Phi}^T\mathbf{W}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^T\mathbf{W}\mathbf{b} . \tag{2.26}$$

Vergleicht man die Schätzung für $\mathbf{c}\left(x^*\right)$ mit (2.20), so stellt man fest, dass $\mathbf{\Phi}^T$ durch $\mathbf{\Phi}^T\mathbf{W}$ substituiert wurde. Mit $\mathbf{W} = \mathbf{I}$ erhält man also das Verfahren der Regression.
Aufgrund des Einführens der Gewichtsmatrix $\mathbf{W}$ muss zur Approximation jeder Auswertungsstelle $\mathbf{x}^*$ der Vektor $\mathbf{c}$ neu bestimmt werden. Dazu kann im Gegensatz zur Regression nicht mehr das Verfahren von HOUSEHOLDER verwendet werden. Anstelle der Berechnung der Inversen sollte jedoch das Gleichungssystem

$$\mathbf{\Phi}^T\mathbf{W}\mathbf{\Phi}\mathbf{c} = \mathbf{\Phi}^T\mathbf{W}\mathbf{b} \tag{2.27}$$

gelöst werden.

2.3 Radiale Basisfunktionen - RBF

Bei mehrdimensionalen Approximationsaufgaben nimmt die Berechnung der Funktionswerte der Basisfunktionen zumeist einen recht großen Anteil ein. Deshalb wird nach Funktionen gesucht, die eine einfache und schnelle Funktionsauswertung ermöglichen.

Ein bekannter Vertreter dieser Klasse von Funktionen sind die radialen Basisfunktionen, die unter anderem in [5] und [7] beschrieben sind.

2.3.1 Definition und Beispiele

Definition 2.5. Eine Funktion $f : \mathbb{R}^m \to \mathbb{R}$ definiert durch

$$f(\mathbf{x}) := \sum_{j=1}^{n} c_j \cdot \phi\left(\|\mathbf{x} - \mathbf{v}^{(j)}\|\right) \tag{2.28}$$

mit $c_j \in \mathbb{R},\ \mathbf{v}^{(j)} \in \mathbb{R}^m$ heißt *radiale Basisfunktion* (RBF).

Bemerkungen.

- Die Vektoren $\mathbf{v}^{(j)}$ werden Zentren genannt.
- Eine Funktion, die allein vom Abstand eines Punktes vom Zentrum abhängt, jedoch nicht von der konkreten Position, heißt radial.
- $\|.\|$ ist eine beliebige Norm, mit $\|.\|_2$ wird die Euklidische Norm bezeichnet.

Um einen Eindruck über die Vielfalt der Gestalt gebräuchlicher radialer Basisfunktionen zu verschaffen, werden einige Beispiele aufgelistet, welche man in ähnlicher Form auch in [7] findet. Dabei wird $r := \|\mathbf{x} - \mathbf{v}^{(j)}\|$ gesetzt.

Radiale Basisfunktionen	$\phi(r)$	Parameter
Gaußfunktionen	$e^{-(kr)^2}$	$k > 0$
Multiquadrics	$\sqrt{r^2 + k^2}$	$k > 0$
Inverse Multiquadrics	$\frac{1}{\sqrt{r^2+k^2}}$	$k > 0$
Inverse Quadrics	$\frac{1}{r^2+k^2}$	$k > 0$
Polyharmonic Splines	r^{2k-1}	$k \in \mathbb{N}$
	$r^{2k} \cdot log(r)$	$k \in \mathbb{N}$

Tabelle 2.1: Beispiele für radiale Basisfunktionen

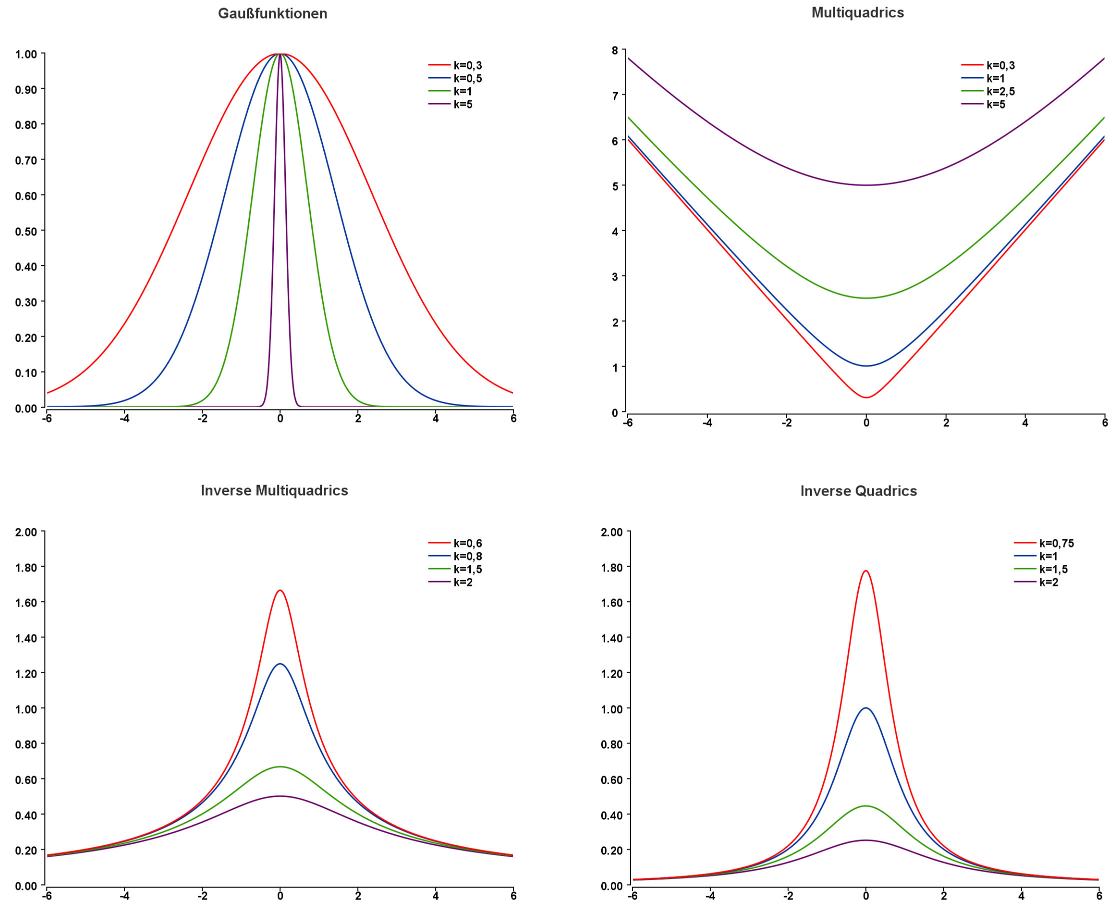

Abbildung 2.2: Graphen radialer Basisfunktionen

2.3.2 RBF-Netze

Werden zur Lösung eines Problems radiale Basisfunktionen genutzt, so verwendet man zumeist den bekanntesten Vertreter dieser Familie von Funktionen, die Gaußfunktionen.

Deshalb wird im Folgenden der Ansatz

$$f\left(\mathbf{x}\right) = \sum_{j=1}^{n} c_j \cdot e^{-\frac{\left\|\mathbf{x}-\mathbf{v}^{(j)}\right\|^2}{2\lambda^2}} \tag{2.29}$$

gewählt.

Dabei bestimmen die Zentren $\mathbf{v}^{(j)}$ die Position der einzelnen Gaußglocken, λ^2 die Steilheit der Flanken und c_j die Gewichtung der Basisfunktionen in der Summenfunktion. Die Anpassung dieser Parameter an ein gestelltes Problem lässt sich mit Hilfe künstlicher neuronaler Netze beschreiben und durchführen.

2.3.2.1 Künstliche neuronale Netze

Künstliche neuronale Netze beziehen sich auf das Neuronennetz im menschlichen Gehirn. In Analogie zur Tätigkeit des Gehirns werden komplexe Probleme gelöst. Ein neuronales Netz lässt sich als gerichteter Graph modellieren, dessen Knoten auch als Neuronen bezeichnet werden und dessen Kanten die Informationsflüsse zwischen den einzelnen Neuronen nachbilden. Um ein neuronales Netz an ein gestelltes Problem anzupassen, wird eine Trainingsmenge verwendet. Der Lernprozess kann durch folgende Aktionen durchgeführt werden (vgl. [10]):

- Verbindungsgewichte ändern,
- Neuronenfunktionen ändern,
- neue Verbindungen entwickeln,
- vorhandene Verbindungen löschen,
- neue Neuronen entwickeln und
- vorhandene Neuronen löschen.

Dabei kann man drei Arten des Lernens unterscheiden. Beim *unüberwachten Lernen* erhält das Netz ausschließlich Beispiele und versucht eigenständig Strukturen in den Daten zu erkennen. Beim *bestärkenden Lernen* erhält das Netz eine Rückmeldung über die Richtigkeit des Ergebnisses, eventuell auch über die Güte der Lösung. Das *überwachte Lernen* ist dadurch gekennzeichnet, dass zu den Trainingsdaten zusätzlich korrekte Lösungen vorhanden sind, mit deren Hilfe das Netz erfährt, welche Abweichungen vom richtigen Wert bei welcher Ausgabe auftraten.

2.3.2.2 RBF-Netze als vollvernetzte Feedforward-Netze

Ein RBF-Netz lässt sich als vollvernetztes Feedforward-Netz analog zu [10] modellieren.

Definition 2.6. Ein *Feedforward-Netz* ist ein neuronales Netz, bei dem die einzelnen Neuronen in Schichten angeordnet sind, wobei die Weitergabe von Informationen stets nur von Neuronen einer Schicht zu Neuronen einer nachfolgenden Schicht erfolgen kann. Es heißt *vollvernetzt*, wenn jede Schicht vollständig mit der folgenden vernetzt ist.

Bemerkungen.

- Ein Feedforward-Netz lässt sich als azyklischer gerichteter Graph modellieren.

- Es sind auch Verbindungen erlaubt, die eine oder mehrere Ebenen überspringen, sogenannte Short-Cut-Verbindungen.
- Bei RBF-Netzen treten keine Short-Cut-Verbindungen auf.

Ein RBF-Netz besteht aus der Eingabeschicht, einer verdeckten Schicht und der Ausgabeschicht. Die Anzahl der Neuronen in der Eingabeschicht entspricht der Dimension des Datenraumes, die der verdeckten Schicht der Anzahl der Zentren. Betrachtet man nur Abbildungen von $\mathbb{R}^m \to \mathbb{R}$, so besteht die Ausgabeschicht entsprechend der Dimension des Zielraumes aus einem Neuron.

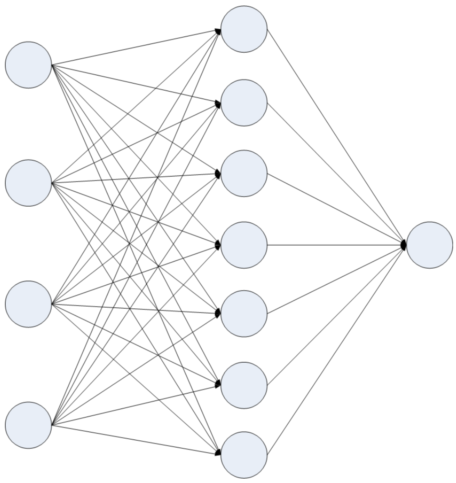

Abbildung 2.3: RBF-Netz

2.3.3 Interpolation und Approximation mit RBF-Netzen

Da ein RBF-Netz nur aus einer verdeckten Schicht besteht, können die Gewichte c_j, wie in [7] beschrieben, direkt berechnet werden. Dies kann sowohl mittels Interpolation als auch Approximation geschehen. Fordert man die Erfüllung der Interpolationsbedingung, so werden die Zentren der Gaußglocken identisch mit den gegebenen Stützstellen gewählt. Dann muss gelten:

$$\sum_{j=1}^{N} c_j \cdot e^{-\frac{\|\mathbf{x}^{(i)}-\mathbf{x}^{(j)}\|^2}{2\lambda^2}} \stackrel{!}{=} y^{(i)},\ \forall i = 1, \ldots, N\ . \tag{2.30}$$

Legt man zudem die Weite λ^2 der Gaußfunktionen fest, so erhält man ein lineares Gleichungssystem zur Bestimmung der Gewichte c_j. Um die Übersichtlichkeit der Darstellung zu gewähren, wird anstelle der Gaußfunktion wieder die allgemeine Darstellung einer radialen Basisfunktion verwendet.

$$\begin{pmatrix} \phi\left(\|\mathbf{x}^{(1)}-\mathbf{x}^{(1)}\|\right) & \phi\left(\|\mathbf{x}^{(1)}-\mathbf{x}^{(2)}\|\right) & \cdots & \phi\left(\|\mathbf{x}^{(1)}-\mathbf{x}^{(N)}\|\right) \\ \phi\left(\|\mathbf{x}^{(2)}-\mathbf{x}^{(1)}\|\right) & \phi\left(\|\mathbf{x}^{(2)}-\mathbf{x}^{(2)}\|\right) & \cdots & \phi\left(\|\mathbf{x}^{(2)}-\mathbf{x}^{(N)}\|\right) \\ \vdots & \vdots & \ddots & \vdots \\ \phi\left(\|\mathbf{x}^{(N)}-\mathbf{x}^{(1)}\|\right) & \phi\left(\|\mathbf{x}^{(N)}-\mathbf{x}^{(2)}\|\right) & \cdots & \phi\left(\|\mathbf{x}^{(N)}-\mathbf{x}^{(N)}\|\right) \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{pmatrix} \tag{2.31}$$

Das entstandene Gleichungsystem lässt sich eindeutig lösen, wenn die symmetrische Systemmatrix regulär ist. Mit wachsender Anzahl der Stützstellen nähert sich die Determinante der Matrix jedoch der Null an, weshalb das Verfahren allgemein nicht praktikabel ist.
Deshalb sollte man zur Approximation übergehen und weniger Zentren als Datenpunkte verwenden. Dabei erhält man, wie bereits in Kapitel 1 dargelegt, eine deutlich glattere Funktion und besitzt zudem mehr Freiheiten in der Wahl der Zentren.
Für die Auswahl der Zentren stehen folgende Verfahren zur Verfügung:

- gleichmäßige Verteilung der Zentren im Raum,
- Verwenden einer willkürlichen Teilmenge der Datenmenge und
- Clustern der Eingangsdaten und Benutzung der Clusterzentren.

Zusätzlich gestaltet sich die Wahl eines geigneten Wertes für λ^2 schwierig, wenn keine Informationen über die Verteilung der Daten bekannt ist. Wurde zur Wahl der Zentren ein Clusteralgorithmus verwendet, so empfiehlt sich die mittlere Abweichung im Cluster, andernfalls könnte man den mittleren Abstand der Zentren verwenden.
Man erhält ein überbestimmtes Gleichungssystem, dessen Normallösung wie in Abschnitt 1.3 mittels Verfahren von HOUSEHOLDER oder über die Pseudoinverse bestimmt werden kann.
Es gibt jedoch viele Gründe, die dagegen sprechen, die Gewichte c_j direkt durch Lösen eines Gleichungssystems zu bestimmen. Zum einen ist das Wahl der Zentren $\mathbf{v}^{(j)}$ und der Steilheit λ^2 zumeist nicht optimal und eine Verschiebung könnte deutlich bessere Ergebnisse erzielen, zum anderen treten bei größeren Systemen oft numerische Instabilitäten auf. Deshalb empfiehlt es sich Lernalgorithmen auf den RBF-Netzen anzuwenden, welche die Zentren, Steilheit und Gewichte iterativ verbessern.

2.3.4 Lernstrategien in RBF-Netzen

Das Ziel des Trainings des RBF-Netzes ist die Minimierung des Fehlers der Netzausgabe. Um die Verbesserungen mit jeder Trainingsiteration messen zu können, wird der summierte quadratische Fehler verwendet. Zunächst jedoch spielt die Bestimmung der

Zentren die übergeordnete Rolle. Dabei wird die Cluster-Variante untersucht und als Verfahren der Neural Gas verwendet.

2.3.4.1 Neural Gas mit Metrikadaption

Der Neural Gas ist ein Cluster-Algorithmus, der auf der Idee eines diffundierenden Gases beruht. Während des Algorithmus wird die von Th. Martinetz aufgestellte Energiefunktion (vgl. [2]), die dem Potenzial eines diffundierenden Gases entspricht, minimiert.

$$E_{NG} = \frac{1}{2C\left(\alpha\right)} \sum_{j=1}^{n} \int P\left(\mathbf{x}\right) h_{\alpha}\left(k_j\left(\mathbf{x}, \mathbf{v}^{(j)}\right)\right) \|\mathbf{x} - \mathbf{v}^{(j)}\| \; d\mathbf{x} \tag{2.32}$$

Dabei beschreibt $P\left(\mathbf{x}\right)$ die Dichte der Daten und h_{α} die Nachbarschaftsfunktion des Neural Gas; $C\left(\alpha\right)$ ist eine Konstante, für die gilt:

$$C\left(\alpha\right) = \sum_{j=1}^{n} h_{\alpha}\left(k_j\right) = \sum_{k=0}^{n-1} h_{\alpha}\left(k\right) \quad . \tag{2.33}$$

Die Nachbarschaftsfunktion des Neural Gas lässt sich darstellen durch:

$$h_{\alpha}\left(k_j\right) = e^{-\frac{k_j(\mathbf{x},\mathbf{v})}{\alpha}} \quad . \tag{2.34}$$

Dabei beschreibt

$$k_j = \left| \left\{ \mathbf{v}^{(q)}, \; \|\mathbf{x} - \mathbf{v}^{(q)}\| < \|\mathbf{x} - \mathbf{v}^{(j)}\| \right\} \right| \tag{2.35}$$

den Rang des j-ten Zentrums bei der Siegerbestimmung.

Beim klassischen Neural Gas wird die Euklidische Metrik als Distanzfunktion verwendet. Mit dieser Metrik kann jedoch nur achsenparallel geclustert werden. Allgemeine Ellipsoiden-Cluster erhält man unter Verwendung der in [2] dargelegten allgemeineren Metrik

$$\|\mathbf{x} - \mathbf{v}^{(j)}\|_{\mathbf{\Lambda}_j} = \left(\mathbf{x} - \mathbf{v}^{(j)}\right)^T \mathbf{\Lambda}_j \left(\mathbf{x} - \mathbf{v}^{(j)}\right) \quad . \tag{2.36}$$

Dabei ist $\mathbf{\Lambda}_j \in \mathbb{R}^{m \times m}$ eine symmetrische, positiv definite Matrix mit der Determinante $\det\left(\mathbf{\Lambda}_j\right) = 1$. Letztere Bedingung garantiert, dass die Matrix bei Minimierung der Energiefunktion des Neural Gas nicht gegen die Nullmatrix konvergiert.

Da $\mathbf{\Lambda}_j$ eine symmetrische, positiv definite Matrix ist, lässt sich diese zerlegen in

$$\mathbf{\Lambda}_j = {\mathbf{\Omega}_j}^T \mathbf{\Omega}_j \; , \tag{2.37}$$

sodass für die Abstandsfunktion gilt:

$$\|\mathbf{x} - \mathbf{v}^{(j)}\|_{\mathbf{\Omega}_j} = \left(\mathbf{x} - \mathbf{v}^{(j)}\right)^T {\mathbf{\Omega}_j}^T \mathbf{\Omega}_j \left(\mathbf{x} - \mathbf{v}^{(j)}\right) \quad . \tag{2.38}$$

Um die Energiefunktion des Neural Gas minimieren zu können, wird der Gradient benötigt. Dazu werden zunächst analog zu [3] die partiellen Ableitungen der Energiefunktion nach den Zentren $\mathbf{v}^{(j)}$ und den Matrizen der Metrik $\mathbf{\Omega}_j$ berechnet.

$$\frac{\partial E_{NG}}{\partial \mathbf{v}^{(j)}} = \frac{1}{2} \cdot e^{-\frac{k\left(\mathbf{x},\mathbf{v}^{(j)}\right)}{\alpha}} \cdot (-2) \cdot {\mathbf{\Omega}_j}^T \mathbf{\Omega}_j \left(\mathbf{x} - \mathbf{w}^{(j)}\right)$$

$$= -e^{-\frac{k\left(\mathbf{x},\mathbf{v}^{(j)}\right)}{\alpha}} \cdot {\mathbf{\Omega}_j}^T \mathbf{\Omega}_j \left(\mathbf{x} - \mathbf{w}^{(j)}\right) \tag{2.39}$$

$$\frac{\partial E_{NG}}{\partial \mathbf{\Omega}_j} = \frac{1}{2} \cdot e^{-\frac{k\left(\mathbf{x},\mathbf{v}^{(j)}\right)}{\alpha}} \cdot \frac{\partial \|\mathbf{x} - \mathbf{v}^{(j)}\|_{\mathbf{\Omega}_j}}{\partial \mathbf{\Omega}_j}$$

$$= \frac{1}{2} \cdot e^{-\frac{k\left(\mathbf{x},\mathbf{v}^{(j)}\right)}{\alpha}} \cdot 2 \cdot \left(\mathbf{x} - \mathbf{v}^{(j)}\right) \left(\mathbf{x} - \mathbf{v}^{(j)}\right)^T {\mathbf{\Omega}_j}^T$$

$$= e^{-\frac{k\left(\mathbf{x},\mathbf{v}^{(j)}\right)}{\alpha}} \cdot \left(\mathbf{x} - \mathbf{v}^{(j)}\right) \left(\mathbf{x} - \mathbf{v}^{(j)}\right)^T {\mathbf{\Omega}_j}^T \tag{2.40}$$

Mit Hilfe der partiellen Ableitungen können nun die Lernregeln formuliert werden. Da die Energiefunktion minimiert werden soll, werden die Parameter jeweils in Richtung des negativen Gradienten verschoben.
Es ergeben sich die folgenden Update-Regeln:

$${\mathbf{v}^{(j)}}_{NEU} = \mathbf{v}^{(j)} - \epsilon_{\mathbf{v}} \cdot \frac{\partial E_{NG}}{\partial \mathbf{v}^{(j)}}$$

$${\mathbf{v}^{(j)}}_{NEU} = \mathbf{v}^{(j)} + \epsilon_{\mathbf{v}} \cdot e^{-\frac{k\left(\mathbf{x},\mathbf{v}^{(j)}\right)}{\alpha}} \cdot {\mathbf{\Omega}_j}^T \mathbf{\Omega}_j \left(\mathbf{x} - \mathbf{v}^{(j)}\right) \tag{2.41}$$

$$\mathbf{\Omega}_{j\,NEU} = \mathbf{\Omega}_j - \epsilon_{\mathbf{\Omega}} \cdot \frac{\partial E_{NG}}{\partial \mathbf{\Omega}_j}$$

$$\mathbf{\Omega}_{j\,NEU} = \mathbf{\Omega}_j - \epsilon_{\mathbf{\Omega}} \cdot e^{-\frac{k\left(\mathbf{x},\mathbf{v}^{(j)}\right)}{\alpha}} \cdot \left(\mathbf{x} - \mathbf{v}^{(j)}\right) \left(\mathbf{x} - \mathbf{v}^{(j)}\right)^T {\mathbf{\Omega}_j}^T \tag{2.42}$$

Nach Durchführung des Updates werden in der Regel die an die Matrix $\mathbf{\Lambda}$ gestellten Forderungen nicht erfüllt sein. Deshalb muss die Matrix $\mathbf{\Omega}$ noch so reskaliert werden, dass die Determinante von $\mathbf{\Lambda}$ gleich Eins ist.

Da $\mathbf{\Lambda} = \mathbf{\Omega}^T\mathbf{\Omega}$, muss gelten:

$$\det\left(\mathbf{\Omega}^T\mathbf{\Omega}\right) = 1$$

$$\det\left(\mathbf{\Omega}^T\right) \cdot \det\left(\mathbf{\Omega}\right) = 1$$

Die Determinante der Transponierten einer Matrix ist gleich der Determinante der Ausgangsmatrix, sodass gilt:

$$\left(\det\left(\mathbf{\Omega}\right)\right)^2 = 1$$

$$\left|\det\left(\mathbf{\Omega}\right)\right| = 1 \tag{2.43}$$

Die Bedingung (2.43) soll erfüllt werden, in dem die Matrizen $\mathbf{\Omega}_{j_{NEU}}$ jeweils mit einem geeigneten skalaren Faktor multipliziert werden.
Sei $\det\left(\mathbf{\Omega}_{j_{NEU}}\right) = k$. Dann gilt nach den Regeln der Determinanten:

$$\det\left(a \cdot \mathbf{\Omega}_{j_{NEU}}\right) = a^m \cdot k \ . \tag{2.44}$$

Somit lässt sich (2.43) erreichen, indem man den Koeffizienten a wie folgt wählt:

$$a = \sqrt[m]{\frac{1}{|k|}} \ . \tag{2.45}$$

Dementsprechend gilt dann für die reskalierte Matrix $\mathbf{\Omega}_{j_{SKAL}}$:

$$\mathbf{\Omega}_{j_{SKAL}} = \sqrt[m]{\frac{1}{\left|\det\left(\mathbf{\Omega}_{j_{NEU}}\right)\right|}} \cdot \mathbf{\Omega}_{j_{NEU}} \ . \tag{2.46}$$

Nachdem die Update-Regeln formuliert wurden, müssen nun noch Anfangswerte definiert und die Nachbarschaftsreichweite α sowie geeignete Schrittweiten $\epsilon_{\mathbf{v}}$ und $\epsilon_{\mathbf{\Omega}}$ gewählt werden. Dabei muss zum einen darauf geachtet werden, dass die Zentren sowie die Metrik zunächst stark an neue Datenpunkte angepasst wird und gegen Ende des Verfahrens sich diese nur noch wenig ändern, zum anderen, dass die Zentren deutlich schneller gelernt werden als die Metrik. Auch die Nachbarschaftsreichweite sollte im Laufe des Verfahrens verkleinert werden, damit nur noch die Zentren in unmittelbarer Nähe des Datenpunktes beeinflusst werden.
In Tabelle 2.2 sind geeignete Werte für die Parameter dargestellt:

Parameter	Startwert	Endwert
α	$\frac{n}{3}$	$0.01 \ \ldots \ 0.1$
$\epsilon_{\mathbf{v}}$	$0.4 \ \ldots \ 0.5$	$0.01 \ \ldots \ 0.05$
$\epsilon_{\mathbf{\Omega}}$	$0.002 \ \ldots \ 0.025$	$5 \cdot 10^{-5} \ \ldots \ 0.0025$

Tabelle 2.2: Wahl der Parameter des Neural Gas

Die Zentren werden zu Beginn zufällig initialisiert, die Matrizen $\mathbf{\Omega}_j$ mit der Einheitsmatrix. Bei der Verwendung des Neural Gas ist darauf zu achten, dass auch die Datenpunkte zufällig verwendet werden, damit diese keiner Ordnung unterliegen. Über mehrere Iterationen werden jeweils alle Datenpunkte in zufälliger Reihenfolge verwendet. Nach jedem Datenpunkt werden die Werte der Nachbarschaftsreichweite und Lernschrittweiten exponentiell verringert.

2.3.4.2 Minimierung des quadratischen Fehlers

Nachdem mit Hilfe des Neural Gas die Zentren sowie deren lokale Metrik bestimmt wurden, beginnt das eigentliche Training des RBF-Netzes. Auch dabei wird wieder eine Energiefunktion aufgestellt, die minimiert werden soll. Da das Training des Netzes darauf ausgerichtet ist, den Netzfehler zu minimieren, wird als Energiefunktion der in [5] beschriebene summierte quadratische Fehler verwendet.

$$
\begin{aligned}
E_{RBF} &= \frac{1}{2} \sum_{i=1}^{N} \left(y^{(i)} - f\left(\mathbf{x}^{(i)}\right) \right)^2 \\
&= \frac{1}{2} \sum_{i=1}^{N} \left(y^{(i)} - \sum_{j=1}^{n} c_j \cdot e^{-\frac{\|\mathbf{x}^{(i)} - \mathbf{v}^{(j)}\|_{\mathbf{\Omega}_j}{}^2}{2\lambda^2}} \right)^2
\end{aligned}
$$

Damit sich das RBF-Netz noch besser an die gegeben Daten anpassen kann, wird nicht ein globales λ^2 gewählt, sondern lokal für jedes Zentrum ein ${\lambda_j}^2$.

$$
E_{RBF} = \frac{1}{2} \sum_{i=1}^{N} \left(y^{(i)} - \sum_{j=1}^{n} c_j \cdot e^{-\frac{\|\mathbf{x}^{(i)} - \mathbf{v}^{(j)}\|_{\mathbf{\Omega}_j}{}^2}{2{\lambda_j}^2}} \right)^2 \tag{2.47}
$$

Analog zum Vorgehen beim Neural Gas werden die partiellen Ableitungen der Energiefunktion nach c_j, $\mathbf{v}^{(j)}$ und λ_j gebildet. Dabei wird für $y^{(i)} - f\left(\mathbf{x}^{(i)}\right)$ die Bezeichung $\Delta y^{(i)}$ verwendet.

$$\frac{\partial E_{RBF}}{\partial c_j} = -\Delta y^{(i)} \cdot e^{-\frac{\|\mathbf{x}^{(i)}-\mathbf{v}^{(j)}\|_{\mathbf{\Omega}_j}{}^2}{2\lambda_j{}^2}} \tag{2.48}$$

$$\frac{\partial E_{RBF}}{\partial \lambda_j} = -\Delta y^{(i)} \cdot e^{-\frac{\|\mathbf{x}^{(i)}-\mathbf{v}^{(j)}\|_{\mathbf{\Omega}_j}{}^2}{2\lambda_j{}^2}} \cdot c_j \cdot \frac{\|\mathbf{x}^{(i)}-\mathbf{v}^{(j)}\|_{\mathbf{\Omega}_j}{}^2}{\lambda_j{}^3} \tag{2.49}$$

$$\frac{\partial E_{RBF}}{\partial \mathbf{v}^{(j)}} = -\Delta y^{(i)} \cdot e^{-\frac{\|\mathbf{x}^{(i)}-\mathbf{v}^{(j)}\|_{\mathbf{\Omega}_j}{}^2}{2\lambda_j{}^2}} \cdot c_j \cdot \frac{1}{\lambda_j{}^2} \cdot \mathbf{\Omega}^T\mathbf{\Omega}\left(\mathbf{x}^{(i)}-\mathbf{v}^{(j)}\right) \tag{2.50}$$

Die Update-Regeln lassen sich wiederum wie folgt definieren:

$$c_{j\,NEU} = c_j - \epsilon_c \cdot \frac{\partial E_{RBF}}{\partial c_j} \;, \tag{2.51}$$

$$\lambda_{j\,NEU} = \lambda_j - \epsilon_\lambda \cdot \frac{\partial E_{RBF}}{\partial \lambda_j} \;, \tag{2.52}$$

$$\mathbf{v}^{(j)}{}_{NEU} = \mathbf{v}^{(j)} - \epsilon_{\mathbf{v}} \cdot \frac{\partial E_{RBF}}{\partial \mathbf{v}^{(j)}} \;. \tag{2.53}$$

Dabei sollten die Gewichte c_j zu Beginn zufällig und klein gewählt werden, z. B. im Intervall $[-1, 1]$, λ_j kann gleich 0.5 gesetzt werden. Bei den Lernschrittweiten ϵ sollte man darauf achten, dass die Schrittweite für die Zentren kleiner gewählt wird, da diese ja bereits mit Hilfe des Neural Gas gelernt wurden. Zudem sollte die Schrittweite ϵ_λ zunächst in ähnlicher Größenordnung gewählt werden wie ϵ_c, jedoch deutlich schneller verringert werden, damit gegen Ende des Algorithmus fast nur noch die Gewichte der Gaussfunktionen gelernt werden. Folgende Schrittweiten sind möglich:

Parameter	Startwert	Endwert
ϵ_c	0.05	0.01
ϵ_λ	0.03	0.0025
$\epsilon_{\mathbf{v}}$	0.01	0.001

Tabelle 2.3: Wahl der Lernschrittweiten in RBF-Netzen

Analog zum Lernen mit Hilfe des Neural Gas werden über mehrere Iterationen jeweils alle Datenpunkte in zufälliger Reihenfolge zum Training verwendet und die Lernschrittweiten exponentiell verringert.

2.4 Verfahren von Shepard

Das Verfahren von Shepard, welches in [4] und [12] untersucht wird, lässt sich in die Kategorie der Approximationsverfahren einordnen, die besonders für Streudaten verwendet werden. Zudem gehört es zur Gruppe der Interpolationsverfahren, weshalb auch der Name Shepard-Interpolation gebräuchlich ist.

2.4.1 Idee des Verfahrens

Die Shepard-Interpolation bildet einen interessanten Spezialfall der radialen Basisfunktionen. Bei dieser Methode wird angenommen, dass die Stützwerte $y^{(i)},\ i = 1, \dots N$, deren Stützstelle $\mathbf{x}^{(i)}$ einen großen Abstand von der zu interpolierenden Stelle hat, nur einen geringen Einfluss gegenüber solchen Stützwerten, deren Stützstelle nahe bei der Auswertungsstelle liegen, besitzt. Für die Interpolationsfunktion $f : \mathbb{R}^m \rightarrow \mathbb{R}$ wählt man den Ansatz:

$$f\left(\mathbf{x}\right) = \sum_{i=0}^{N} w_i\left(\mathbf{x}\right) y^{(i)} \ . \tag{2.54}$$

Dabei sind w_i Gewichte mit $w_i\left(\mathbf{x}\right) \geq 0,\ \forall i = 1, \dots N$ und $\sum\limits_{i=1}^{N} w_i\left(\mathbf{x}\right) = 1$.
Um die N Interpolationsbedingungen

$$y^{(k)} = f\left(\mathbf{x}^{(k)}\right) = \sum_{i=0}^{N} w_i\left(\mathbf{x}^{(k)}\right) y^{(i)},\ k = 1, \dots, N \tag{2.55}$$

zu gewährleisten, werden an Stützstellen die Gewichte wie folgt gewählt:

$$w_i\left(\mathbf{x}^{(k)}\right) = \delta_{ki} \ . \tag{2.56}$$

Die Gewichte $w_i\left(\mathbf{x}\right)$ müssen nun noch für eine Stelle $\mathbf{x}^*$ bestimmt werden, für die $\mathbf{x}^* \neq \mathbf{x}^{(i)}$ für $i = 1, \dots, N$ gilt. Die Gewichte werden mit Hilfe des euklidischen Abstandes

$$r_i\left(\mathbf{x}^*\right) = \|\mathbf{x}^* - \mathbf{x}^{(i)}\| \tag{2.57}$$

und einem Parameter $\beta,\ 0 < \beta < \infty$ definiert durch

$$\varphi_i\left(\mathbf{x}^*\right) = \frac{1}{r_i\left(\mathbf{x}^*\right)} \tag{2.58}$$

$$w_i\left(\mathbf{x}^*\right) = \frac{{\varphi_i}^{\beta}}{\sum\limits_{k=1}^{N} {\varphi_k}^{\beta}},\ i = 1, \dots, N \ . \tag{2.59}$$

Damit sind die an die Gewichte gestellten Forderungen der Nichtnegativität und Normiertheit erfüllt.
Für den Ansatz (2.54) ergibt sich mit (2.59) die Darstellung

$$f\left(\mathbf{x}^*\right) = \frac{\sum\limits_{i=0}^{N} {\varphi_i}^{\beta} y^{(i)}}{\sum\limits_{k=0}^{N} {\varphi_k}^{\beta}} \;. \tag{2.60}$$

Der Exponent β in (2.59) ist frei wählbar und wird auch als Glättungsparameter bezeichnet.
Der folgende Algorithmus liefert zu einer Interpolationsstelle $\mathbf{x}^*$ den Funktionswert $f\left(\mathbf{x}^*\right)$.

Algorithmus 2.2 Globale Shepard-Interpolation

`Gegeben` : N Interpolationsstellen $\left(\mathbf{x}^{(i)}, y^{(i)}\right)$, $\beta > 0$, Auswertungsstelle $\mathbf{x}^*$
`Gesucht` : Wert $f\left(\mathbf{x}^*\right)$ der globalen Shepard-Funktion

```
Y = 0, S = 0;
for i = 1 → N do
    r = ‖x* − x^(i)‖;
    if r > 0 then
        w = r^(−β);
        S = S + w;
        Y = Y + w · y^(i);
    end
    else if r = 0 then
        S = 1;
        Y = y^(i);
        break;
    end
end
f(x*) = Y/S;
```

2.4.2 Lokale Shepard-Interpolation

In [4] wird vorgeschlagen, zur Berechnung eines Funktionswertes $f\left(\mathbf{x}^*\right)$ nicht alle N Stützstellen zu verwenden. Bei dieser lokalen Variante der Shepard-Interpolation werden nur die M Stellen $\mathbf{x}^{(i)}$ verwendet, die innerhalb einer m-dimensionalen Kugel mit dem Radius R um die Auswertungsstelle liegen. Alle anderen Stützstellen erhalten das Gewicht $w_i\left(\mathbf{x}^*\right) = 0$. Wie in [4] empfohlen, werden die sogenannten Franke-Little-Gewichte verwendet.

Mit

$$\varphi_i(\mathbf{x}^*) = 1 - \frac{r_i(\mathbf{x}^*)}{R} \text{ für } 0 < r_i(\mathbf{x}^*) < R \tag{2.61}$$

ergeben sich die Gewichte der lokalen Shepard-Funktion:

$$w_i(\mathbf{x}^*) = \frac{{\varphi_i}^\beta}{\sum\limits_{k=1}^{N} {\varphi_k}^\beta},\ i = 1, \ldots, M\ . \tag{2.62}$$

Der folgende Algorithmus liefert zu einer Interpolationsstelle $\mathbf{x}^*$ den Wert der lokalen Shepard-Funktion $f(\mathbf{x}^*)$.

Algorithmus 2.3 Lokale Shepard-Interpolation

`Gegeben` : N Interpolationsstellen $\left(\mathbf{x}^{(i)}, y^{(i)}\right)$, $\beta > 0$, $R > 0$, Auswertungsstelle $\mathbf{x}^*$
`Gesucht` : Wert $f(\mathbf{x}^*)$ der globalen Shepard-Funktion

$Y = 0, S = 0$;
for $i = 1 \to N$ **do**
  $r = \|\mathbf{x}^* - \mathbf{x}^{(i)}\|$;
  if $0 < r < R$ **then**
    $w = \left(1 - \frac{r}{R}\right)^\beta$;
    $S = S + w$;
    $Y = Y + w \cdot y^{(i)}$;
  end
  else if $r \geq R$ **then**
    continue;
  end
  else if $r = 0$ **then**
    $S = 1$;
    $Y = y^{(i)}$;
    break;
  end
end
if $S > 0$ **then**
  $f(\mathbf{x}^*) = \frac{Y}{S}$;
end
else if $S = 0$ **then**
  Im Abstand R von $\mathbf{x}^*$ liegen keine Stützstellen, R muss vergrößert werden.
end

2.5 Krigingverfahren

Die Approximationsmethode, die unter dem Namen Kriging bekannt ist, wurde vom Bauingenieur Daniel Gerhardus Krige im Jahre 1951 eingeführt. In den sechziger Jahren des zwanzigsten Jahrhunderts wurden die Arbeiten Kriges durch den französischen

Mathematiker Georges Matheron um den mathematischen Hintergrund erweitert. Mit diesen Arbeiten entwickelte sich die Geostatik, ein Teilgebiet der Geologie. Trotz des Alters des Verfahrens gehört es noch zum aktuellen Forschungsbestand, da insbesondere die geeignete Bestimmung der Parameter nicht hundertprozentig geklärt ist.

2.5.1 Verfahrensidee

Beim Kriging-Verfahren wird angenommen, dass sich die Approximationsfunktion für jeden Punkt $\left(\mathbf{x}^{(i)}, y^{(i)}\right),\ i = 1, \ldots, N$ darstellen lässt als:

$$\begin{aligned} \text{Funktionswert} \quad &= \quad \text{linearer Anteil} \quad + \quad \text{systematischer Fehler} \\ y^{(i)} \quad &= \quad \sum_{j=1}^{m} c_j \cdot \phi^{(j)}(\mathbf{x}) \quad + \quad \varepsilon(\mathbf{x}) \ . \end{aligned} \tag{2.63}$$

In diesem Modell stellen $\phi^{(j)} : \mathbb{R}^m \to \mathbb{R},\ j = 1, \ldots, m$ lineare Funktionen und $c_j \in \mathbb{R},\ j = 1, \ldots, m$ unbekannte Koeffizienten dar, die noch aus den gegebenen Daten bestimmt werden müssen. Der Fehler, der durch Approximation einer nichtlinearen Funktion mittels linearen Basisfunktionen auftritt wird durch den Term $\varepsilon : \mathbb{R}^m \to \mathbb{R}$ beschrieben. In der Wahl der linearen Basisfunktion liegen die größten Freiheiten des Kriging-Verfahrens. Folgende drei grundlegenden Ansätze werden in der Literatur genannt (vgl. [14] und [15]):

- der *einfache Kriging*, bei dem angenommen wird, dass das lineare Modell identisch null ist, d.h.:

$$y^{(i)} = \hat{f}\left(\mathbf{x}^{(i)}\right) = \varepsilon\left(\mathbf{x}^{(i)}\right),\ i = 1, \ldots, N \ . \tag{2.64}$$

- der *gewöhnliche Kriging*, bei dem das lineare Modell als konstanter Verlauf um einen Mittelwert $\mu \in \mathbb{R}$ angenommen wird:

$$y^{(i)} = \hat{f}\left(\mathbf{x}^{(i)}\right) = \mu + \varepsilon\left(\mathbf{x}^{(i)}\right),\ i = 1, \ldots, N,\ \mu \in \mathbb{R} \ . \tag{2.65}$$

- der *allgemeine Kriging*, bei dem das lineare Modell aus (2.63) verwendet wird.

Durch den Einsatz des allgemeinen Kriging können wahrscheinlich die besten Approximationsergebnisse erzielt werden. Jedoch lassen sich die linearen Basisfunktionen aufgrund fehlender Informationen über die Verteilung der Datenmenge nur schwer ermitteln. Deshalb wird im Folgenden, ähnlich wie in [14] und [15] der gewöhnliche Kriging untersucht.

Durch die Vereinfachung des Kriging-Modells muss deutlich mehr Wert auf den Fehlerterm gelegt werden. Man nimmt an, dass die Fehlerterme normalverteilt mit dem

Erwartungswert null und der Varianz σ^2 sind. Zudem wird davon ausgegangen, dass sich die Fehlerterme für benachbarte Punkte ähnlich verhalten, während bei Punkten mit großem Abstand diese sehr stark variieren können. Dann besteht zwischen der Korrelation der Fehlerterme und dem Abstand der Punkte ein enger Zusammenhang. Die Berechnung des Abstandes d zweier Punkte kann mit verschiedensten Abstandsmaßen erfolgen; beim Kriging wird jedoch gewöhnlich die folgende Distanzfunktion verwendet:

$$d : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}_0^+ : (\hat{\mathbf{x}}, \bar{\mathbf{x}}) \mapsto \sum_{k=1}^{m} \theta_k |\hat{\mathbf{x}}_k - \bar{\mathbf{x}}_k|^{p_k} \; , \tag{2.66}$$

wobei $\theta_k > 0,\ p_k \in [1,2]\,,\ k = 1, \ldots, m$. Mit Hilfe der θ_k kann der Einflussbereich jedes Punktes verändert werden. Bei kleinem θ_k hat jeder Punkt einen großen Einflussbereich, während bei großem θ_k nur die Fehler der Punkte in der unmittelbaren Umgebung eines gegebenen $\mathbf{x}$ stark korreliert. Die Parameter p_k bestimmen die Glattheit des Modells. Wählt man alle $p_k = 2$, so ist die Distanzfunktion unendlich oft differenzierbar. Da die Modellparameter für jede Dimension bestimmt werden, kann das Verhalten der Funktion in den einzelnen Raumrichtungen recht schnell erkannt werden.
Mit Hilfe des Abstandsmaßes lässt sich nun die Korrelation $\mathbf{Corr} := \mathbf{Corr}\left[\varepsilon\left(\hat{\mathbf{x}}\right), \varepsilon\left(\bar{\mathbf{x}}\right)\right]$ zweier Punkte $\hat{\mathbf{x}}, \bar{\mathbf{x}}$ berechnen:

$$\mathbf{Corr} : \mathbb{R}^m \times \mathbb{R}^m \to (0,1] : (\hat{\mathbf{x}}, \bar{\mathbf{x}}) \mapsto e^{-d(\hat{\mathbf{x}}, \bar{\mathbf{x}})} \; . \tag{2.67}$$

Die Korrelationen der Fehlerterme aller gegebenen Punkte werden in der Korrelationsmatrix $\mathbf{C}$ zusammengefasst:

$$\mathbf{C} := \left[c_{ij}\right]_{i,j=1}^{N} := \left[\mathbf{Corr}\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right)\right]_{i,j=1}^{N} \; . \tag{2.68}$$

2.5.2 Schätzung der Modellparameter

Zur vollständigen Definition des Kriging-Modells müssen die $2m+2$ Parameter $\mu, \sigma^2, \theta = \left(\theta_1, \ldots, \theta_m\right)^T, \mathbf{p} = \left(p_1, \ldots, p_m\right)^T$ bestimmt werden. Dafür wird ein Maß benötigt, welches die Güte der Approximation bewertet. Auf Grundlage der Überlegungen im Abschnitt 2.5.1 wird die m-dimensionale Normalverteilung verwendet und die Likelihood-Funktion $\mathbf{L} : \mathbb{R} \times \mathbb{R}_+ \times \mathbb{R}_+^m \times [1,2]^m \to (0,\infty)$ definiert:

$$\mathbf{L}\left(\mu, \sigma^2, \theta, \mathbf{p}\right) = \frac{1}{\left(2\pi\sigma^2\right)^{\frac{N}{2}} \sqrt{\det\left(\mathbf{C}\right)}} \exp\left(-\frac{\left(\mathbf{y} - \mu\mathbf{1}\right)^T \mathbf{C}^{-1} \left(\mathbf{y} - \mu\mathbf{1}\right)}{2\sigma^2}\right) \; . \tag{2.69}$$

Je größer der Wert von $\mathbf{L}$, desto besser passt sich das Modell den gegebenen Daten an. Da die Parameter θ und $\mathbf{p}$ nur für das Aufstellen der Korrelationsmatrix

$\mathbf{C}$ benötigt werden, können μ und σ^2 unabhängig von diesen Parametern mittels Maximum-Likelihood-Schätzung bestimmt werden. Durch partielle Differentiation von (2.69) ergeben sich die Schätzwerte:

$$\hat{\mu} = \frac{\mathbf{1}\mathbf{C}^{-1}\mathbf{y}}{\mathbf{1}\mathbf{C}^{-1}\mathbf{1}} \,, \tag{2.70}$$

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mu\mathbf{1})^T \mathbf{C}^{-1} (\mathbf{y} - \mu\mathbf{1})}{N} \,. \tag{2.71}$$

Durch Einsetzen der Schätzwerte für μ und σ^2 in (2.69) erhält man die folgende Beziehung zur Bewertung des Modells:

$$\hat{\mathbf{L}}(\theta, \mathbf{p}) = \left[\left(\frac{2\pi e}{N} \left(\mathbf{y}^T\mathbf{C}^{-1}\mathbf{y} - \frac{\left(\mathbf{1}^T\mathbf{C}^{-1}\mathbf{y}\right)^2}{\mathbf{1}^T\mathbf{C}^{-1}\mathbf{1}} \right) \right)^N \det(\mathbf{C}) \right]^{-\frac{1}{2}} \,. \tag{2.72}$$

Unter Verwendung von $\hat{\mathbf{L}}$ können die Parameter θ und $\mathbf{p}$ optimiert werden, wobei die Güte des Modells mit $\hat{\mathbf{L}}$ wächst. Die Aufgabe

$$\begin{aligned} &\hat{\mathbf{L}}(\theta, \mathbf{p}) \rightarrow \operatorname*{Max}_{\theta, \mathbf{p}} \\ &\theta_k \geq 0 \\ &1 \leq p_k \leq 2 \qquad k = 1, \ldots, m \end{aligned} \tag{2.73}$$

kann mit einem beliebigen Verfahren der nichtlinearen Optimierung gelöst werden.

2.5.3 Schätzung eines unbekannten Funktionswertes

Nachdem mit der Bestimmung der Parameter das Kriging-Modell aus Abschnitt 2.5.1 vollständig definiert wurde, kann nun der Funktionswert eines neuen Punktes $\mathbf{x}^* \in \mathbb{R}^m$ geschätzt werden. Dazu wird zunächst die Korrelation $\mathbf{r}$ von $\mathbf{x}^*$ mit den Modellpunkten $\mathbf{x}^{(i)}$ für $i = 1, \ldots, n$ berechnet:

$$\begin{aligned} \mathbf{r} &= (r_1, \ldots, r_N)^T \\ r_i &= \mathbf{Corr}\left(\mathbf{x}^{(i)}, \mathbf{x}^*\right), \; i = 1, \ldots, N \,. \end{aligned} \tag{2.74}$$

Der geschätzte Funktionswert $\hat{f} : \mathbb{R}^m \rightarrow \mathbb{R}$, ausführlich hergeleitet in [14], ergibt sich dann aus:

$$\hat{f}(\mathbf{x}^*) = \hat{\mu} + \mathbf{r}^T\mathbf{C}^{-1}(\mathbf{y} - \hat{\mu}\mathbf{1}) \,. \tag{2.75}$$

Dabei wirkt der Schätzer $\hat{f}(\mathbf{x}^*)$ interpolierend, was in [14] bewiesen wurde.

3 Modellgüte von Ersatzmodellen

3.1 Mathematische Ersatzmodelle

Die Optimierung von komplexen Bauteilen im Antriebsstrang eines Fahrzeugs erfordert zunehmend die Verwendung von mathematischen Ersatzmodellen. Dabei werden mit Hilfe von Approximationsmethoden die aufwändigen Zielfunktionen nachgebildet, um dann unter Verwendung des Modells erfolgversprechende Punkte für die Optimierung zu suchen und an diesen eine Zielfunktionsauswertung durchzuführen. Anschließend werden die neuen Punkte ins Ersatzmodell aufgenommen und das Verfahren fortgesetzt. Dieses Vorgehen eignet sich besonders für Optimierungsaufgaben, bei denen die Anzahl der Zielfunktionsauswertungen sehr stark begrenzt ist. (vgl. [15])
Zum einen muss die Zielfunktion mit einer begrenzten Anzahl von Datenpunkten ohne Gradienteninformationen modelliert werden können, zum anderen sollte das Modell in der Lage sein, die Zielfunktion hinreichend genau zu approximieren. Um die Güte der Approximation beurteilen zu können, müssen Bewertungskriterien für das Modell entwickelt werden.

3.2 Globale Bewertungskriterien für die Güte eines Ersatzmodells

Mit Hilfe globaler Bewertungskriterien soll beurteilt werden, wie gut sich das Modell an die gegebenen Daten anpasst. Bei der Auswahl der Kriterien ist darauf zu achten, dass sich die Ergebnisse für verschiedene Approximationsverfahren vergleichen lassen. Da bei interpolierenden Verfahren, wie der Shepard-Interpolation und dem Krigingverfahren, für alle Datenpunkte die Funktionswerte des Modells mit den exakten Funktionswerten an den Stützstellen übereinstimmen, würden die Modelle, die mit diesen Verfahren erstellt wurden, stets optimale Bewertungsergebnisse erzielen. Deshalb wird die gegebene Datenmenge in eine Trainings- und eine Testmenge unterteilt. Die Trainingsmenge wird verwendet, um das Modell aufzubauen, mit Hilfe der Testmenge wird die Güte des Modells bewertet. Dazu können folgende statistische Kenngrößen verwendet werden:

Kenngröße	Symbol	Berechnungsformel
maximaler absoluter Betragsfehler	Δ_{Max}	$\underset{i=1,\ldots,K}{\mathrm{Max}} \; \lvert y^{(i)} - \hat{y}^{(i)} \rvert$
mittlerer absoluter Betragsfehler	Δ_{Mean}	$\frac{1}{K} \cdot \sum\limits_{i=1}^{K} \lvert y^{(i)} - \hat{y}^{(i)} \rvert$
maximaler relativer Betragsfehler	ε_{Max}	$\underset{i=1,\ldots,K}{\mathrm{Max}} \; \lvert \frac{y^{(i)} - \hat{y}^{(i)}}{y^{(i)}} \rvert$
mittlerer relativer Betragsfehler	ε_{Mean}	$\frac{1}{K} \cdot \sum\limits_{i=1}^{K} \lvert \frac{y^{(i)} - \hat{y}^{(i)}}{y^{(i)}} \rvert$
mittlerer quadratischer Fehler	Δ_{MQ}	$\frac{1}{K} \cdot \sum\limits_{i=1}^{K} \left(y^{(i)} - \hat{y}^{(i)} \right)^2$
Bestimmtheitsmaß	b	$\frac{\sum_{i=1}^{K} \left(y^{(i)} - \bar{y} \right)^2 - \sum_{i=1}^{K} \left(y^{(i)} - \hat{y}^{(i)} \right)^2}{\sum_{i=1}^{K} \left(y^{(i)} - \bar{y} \right)^2}$

Tabelle 3.1: Statistische Kenngrößen

Dabei bezeichnet $y^{(i)}$ den exakten Funktionswert, $\hat{y}^{(i)}$ den geschätzten Wert und $\bar{y}$ das arithmetische Mittel der exakten Funktionswerte der Testpunkte.
Bei den unterschiedlichen Fehlerarten wird jeweils ein Wert nahe null angestrebt. Das Bestimmtheitsmaß hingegen nimmt Werte ≤ 1 an und ist ein Maß dafür, wie gut die Approximationsfunktion den Trend der Punktewolke erfasst. Es wird ursprünglich für die lineare Regression in dieser Art und Weise definiert und nimmt dabei Werte zwischen null und eins an. Liegt der Wert des Bestimmtheitsmaßes nahe eins, so sind die Messpunkte nah um die Approximationsfunktion gruppiert, während bei einem Wert deutlich kleiner als eins der Trend kaum erfasst wird.
Bei der Definition des Bestimmtheitsmaßes wird jedoch davon ausgegangen, dass dieses mit den gleichen Punkten berechnet wird, die am Aufbau des Modells beteiligt sind. Verwendet man eine Testmenge zur Berechnung des Bestimmtheitsmaßes, so kann dieses auch Werte kleiner null annehmen.

Obwohl das Bestimmtheitsmaß nun nicht mehr auf das Intervall $[0; 1]$ normiert ist, ermöglicht es gute Vergleiche der einzelnen Approximationsverfahren. Es treten jedoch unerwünschte Effekte bei Funktionen mit größeren Gradienten auf. Ist die Approximationsfunktion nur leicht verschoben gegenüber der unbekannten Ausgangsfunktion, so werden an Stellen mit großen Gradienten sehr große Fehler mithilfe der Testmenge berechnet und eine „flach“ verlaufende Funktion im Vergleich dazu viel zu gut bewertet. (vgl. Abbildung 3.1)

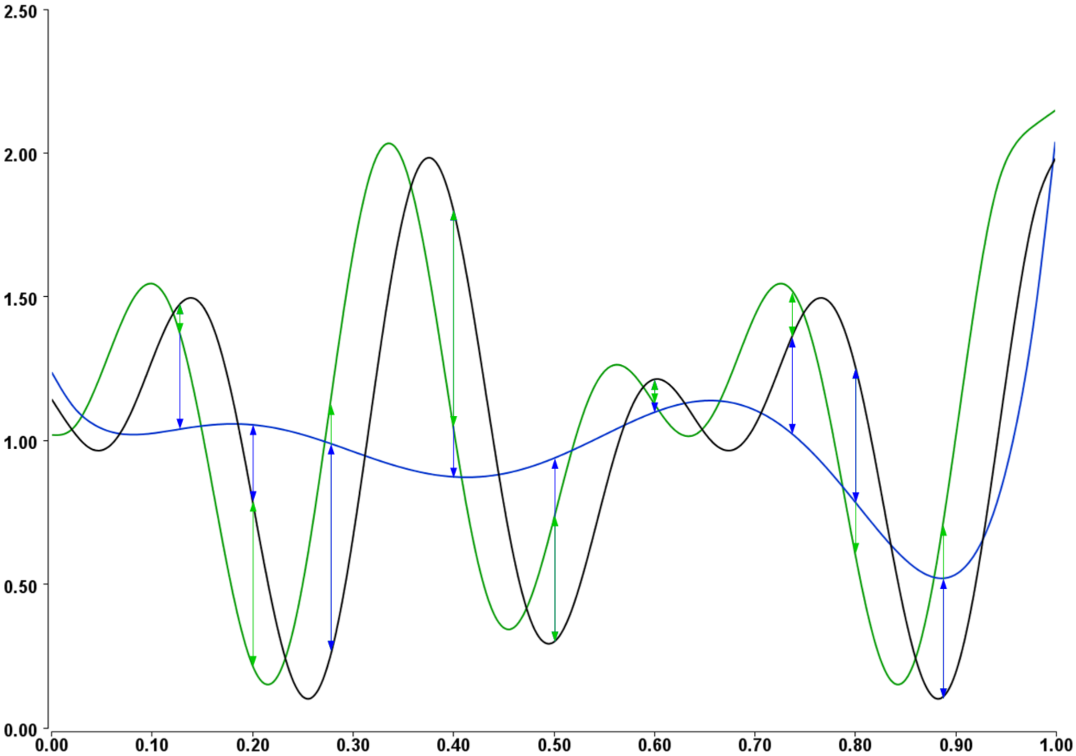

Abbildung 3.1: Vergleich der ermittelten Fehler zweier eindimensionaler Approximationsfunktionen

In Abbildung 3.1 beschreibt die schwarz dargestellte Kurve die Ausgangsfunktion und die beiden anderen Kurven Schätzungen dieser. Berechnet man die Bestimmtheitsmaße der beiden Approximationsfunktionen, so ergibt sich für die grün dargestellte Kurve, welche gegenüber der Ausgangsfunktion nur leicht verschoben wurde, ein Wert von 0,05 und für die blau dargestellte Kurve, die deutlich zu flach verläuft ein Wert von 0,2. Diese Ergebnisse erwecken den Anschein, dass die blau dargestellte Kurve die deutlich besseren Approximationseigenschaften besitzt.

Um zu verhindern, dass Verschiebungen der Approximationsfunktion gegenüber der Ausgangsfunktion zu einer starken Verminderung der ermittelten Güte der Approximation führen, soll im Folgenden der Normalenabstand anstelle des vertikalen absoluten Fehlers verwendet werden. (vgl. Abbildung 3.2)

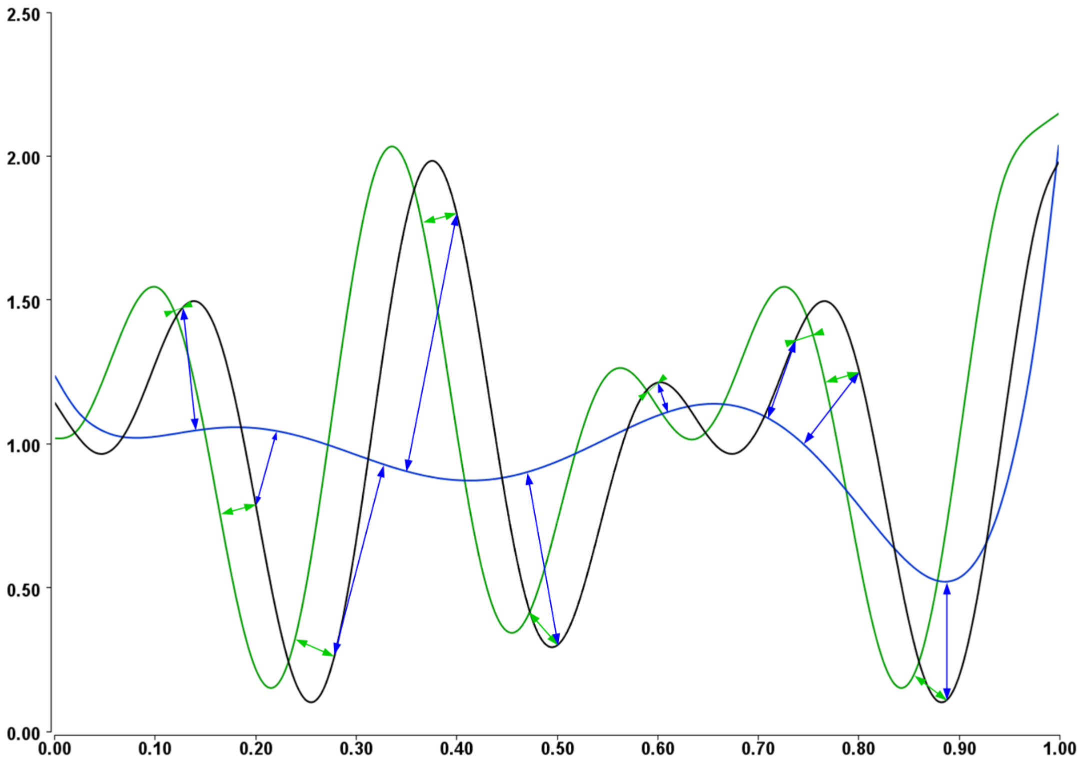

Abbildung 3.2: Normaler Abstand der eindimensionalen Approximationsfunktion von der unbekannten Ausgangsfunktion

In Abbildung 3.2 erkennt man deutlich, dass der Fehler bei der grünen Kurve stark verringert wird, während sich dieser bei der blauen Kurve nur geringfügig ändert. Somit wird wie gewünscht die grüne Kurve unter Verwendung des Normalenabstandes besser bewertet als die blaue Kurve.

Bei praktischen Anwendungen gestaltet sich die Berechnung des normalen Abstandes der Funktionen jedoch recht schwierig. Zum einen sind von der Ausgangsfunktion nur einzelnen Punkte bekannte, sodass nicht der Abstand zweier Funktionen bestimmt werden muss, sondern der normale Abstand eines Punktes zur Approximationsfunktion. Zudem können zumeist mehrere Normalen vom Testpunkt auf die Approximationshyperfläche konstruiert werden.
Mit dem folgenden Verfahren kann ein normaler Abstand bestimmt werden, wobei der Testpunkt mit $(\mathbf{x}^*, y^*)$ gekennzeichnet wird, der Funktionswert der Approximationsfunktion an der Stelle $\mathbf{x}^k$ mit $\hat{y}^k$ und d_{Min} dem minimalen Abstand entspricht.

1. Setze $\mathbf{x}^{(0)} = \mathbf{x}^*$, $d_{Min} = \infty$ und $k = 0$.
2. Berechne $\hat{y}^{(k)}$ mit dem Modell und den Abstand d von $(\mathbf{x}^*, y^*)$ und $\left(\mathbf{x}^{(k)}, \hat{y}^{(k)}\right)$.
3. Ist $d < d_{Min}$, so setze $d_{Min} = d$.

4. Berechne numerisch den Gradienten der Funktion im Punkt $\left(\mathbf{x}^{(k)}, \hat{y}^{(k)}\right)$ und damit die Gleichung der Tangentialhyperebene.

5. Ermittle mit Hilfe der Hesseschen Normalform die orthogonale Projektion des Testpunktes in die Hyperebene, den Lotfußpunkt $(\mathbf{x}_L, y_L)$.

6. Das Verfahren wird abgebrochen, wenn der Abstand des Lotfußpunktes vom aktuellen Punkt $\left(\mathbf{x}^{(k)}, \hat{y}^{(k)}\right)$ kleiner als eine Toleranzgrenze ist.

7. Setze $k = k + 1$, $\mathbf{x}^{(k)} = \mathbf{x}_L$ und führe mit 2. fort.

Mit diesem Algorithmus lässt sich jedoch nur ein Normalenabstand ermittelten. Um den minimalen Abstand eines Testpunktes von der Approximationsfunktion zu erhalten, sollte man die Normalitätsbedingung vernachlässigen und kann dann das Problem als Optimierungsaufgabe mit einer Zielfunktion ohne Nebenbedingungen formulieren. Zur Bestimmung der Lösung dieser Optimierungsaufgabe sollte ein Verfahren verwendet werden, welches keine Gradienten benötigt und eine geringe Anzahl an Zielfunktionsauswertungen ausreicht.
Die Optimierungsaufgabe lautet dann

$$d\left(\mathbf{x}\right) = \left\|\mathbf{x} - \mathbf{x}^*\right\|_2{}^2 + \left(y^* - f\left(\mathbf{x}\right)\right)^2 \rightarrow \underset{\mathbf{x}}{\mathrm{Min}}, \tag{3.1}$$

wobei $f\left(\mathbf{x}\right)$ dem Wert der Approximationsfunktion an der Stelle $\mathbf{x}$ und der Zielfunktionswert dem quadratischen euklidischen Abstand entspricht.
Bei einem iterativen Verfahren sollte $\mathbf{x}^* = (x_1^*, \ldots, x_m^*)$ als Startlösung verwendet werden. Zudem kann der Bereich der Lösung unter Verwendung der Startlösung eingeschränkt werden, weil für die Lösung gelten muss:

$$|x_j - x_j^*| \leq |f\left(\mathbf{x}^*\right) - y^*| \ , \ \forall j = 1, \ldots, m. \tag{3.2}$$

Mit dem Normalenabstand kann nun ein modifiziertes Bestimmtheitsmaß b_{Mod} eingeführt werden, welches an Stelle der Residuen $y - \hat{y}$ den Normalenabstand d_N verwendet. Dieses lässt sich dann wie folgt berechnen:

$$b_{Mod} = 1 - \frac{\sum\limits_{i=1}^{K} d_N\left(\mathbf{x}^{(i)}, y^{(i)}\right)^2}{\sum\limits_{i=1}^{K} \left(y^{(i)} - \bar{y}\right)^2} \tag{3.3}$$

3.3 Lokale Güte der Approximation

Werden mathematische Ersatzmodelle für Optimierungsaufgaben verwendet, so ist es sehr entscheidend, für neue Punkte (im Folgenden Approximationspunkte genannt) des Modells nicht nur eine Schätzung des Funktionswertes zu erhalten, sondern auch eine Abschätzung des Fehlers des Modells, um bei der Suche nach extremalen Punkten eine gewisse Sicherheit zu erhalten. Dafür stehen nur sehr wenige Informationen zur Verfügung. Es lassen sich die absoluten Fehler des Modells an allen Trainings- und Testpunkten bestimmen. In Analogie zur Shepard-Interpolation liegt es nahe, anzunehmen, dass der Einfluss der Punkte mit geringem Abstand zur Approximationsstelle deutlich größer ist als der von Punkten mit großer Distanz zum Approximationspunkt. Deshalb erscheint es nicht sinnvoll alle bekannten Punkte zu verwenden, sondern nur solche mit einem großen Einfluss.

Die Schätzung des Fehlers des Modells kann mit Hilfe der Methode der k nächsten Nachbarn des Approximationspunktes erfolgen, da bei der Festlegung einer Umgebung mit dem Radius R um die Approximationsstelle die Anzahl der Testpunkte innerhalb dieser stark schwanken kann und im Extremfall R sehr oft vergrößert werden muss, damit sich mindestens ein Testpunkt im Inneren befindet.
Da auch die Schätzung der lokalen Güte für alle Approximationsverfahren in gleicher Art und Weise erfolgen soll, muss wieder darauf geachtet werden, dass bei interpolierenden Verfahren der absolute Fehler an Trainingspunkten gleich null ist. Um zu vermeiden, dass diese das Ergebnis verfälschen bzw. sich unter den k nächsten Nachbarn nur Trainingspunkte befinden und damit auch an der Approximationsstelle der Fehler mit null geschätzt wird, werden nur Testpunkte verwendet.
Liegt der Approximationspunkt in einer Epsilon-Umgebung (vgl. Abbildung 3.3) eines Testpunktes, so ist es sinnvoller, den absoluten Fehler des Testpunktes zu übernehmen, da die Mittelung der k Nachbarn zu deutlich größeren Ungenauigkeiten führen kann.

Abbildung 3.3: Epsilon-Umgebung

Um die Genauigkeit der Schätzung des lokalen Approximationsfehlers zu verbessern, empfiehlt es sich die absoluten Fehler der Nachbarn mit dem normierten inversen Abstand von der Approximationsstelle zu gewichten. Der Algorithmus 3.1 liefert die Schätzung des absoluten Fehlers an einem Approximationspunkt.

Algorithmus 3.1 Schätzung des absoluten Fehlers an einer Approximationsstelle

`Gegeben` : K Testpunkte $\left(\mathbf{x}^{(i)}, y^{(i)}\right)$, Approximationsstelle $\mathbf{x}^*, \epsilon$
`Gesucht` : Schätzung $\Delta(\mathbf{x}^*)$ für den absoluten Fehler an der Approximationsstelle
for $i = 1 \rightarrow K$ **do**
  $r_i = \|\mathbf{x}^* - \mathbf{x}^{(i)}\|;$
  if $r_i < \epsilon$ **then**
    $\Delta(\mathbf{x}^*) = y^{(i)} - \hat{y}^{(i)};$
    return;
  end
end
Wähle die k Testpunkte $\mathbf{x}^{(j)}$, für die r_j am kleinsten ist.
Berechne die Summe der inversen Abstände: dSum=$\sum\limits_{j=1}^{k} \frac{1}{r_j}$.
for $j = 1 \rightarrow k$ **do**
  $\Delta\left(\mathbf{x}^{(j)}\right) = y^{(j)} - \hat{y}^{(j)};$
end
$\Delta(\mathbf{x}^*) = \sum\limits_{j=1}^{k} \frac{r_j{}^{-1}}{\text{dSum}} \cdot \Delta\left(\mathbf{x}^{(j)}\right);$

Beispiel 3.1. Anhand von fünf nächsten Nachbarn eines Approximationpunktes wird eine Schätzung des absoluten Fehlers berechnet. Dabei wird mit dem normierten inversen Abstand gewichtet.

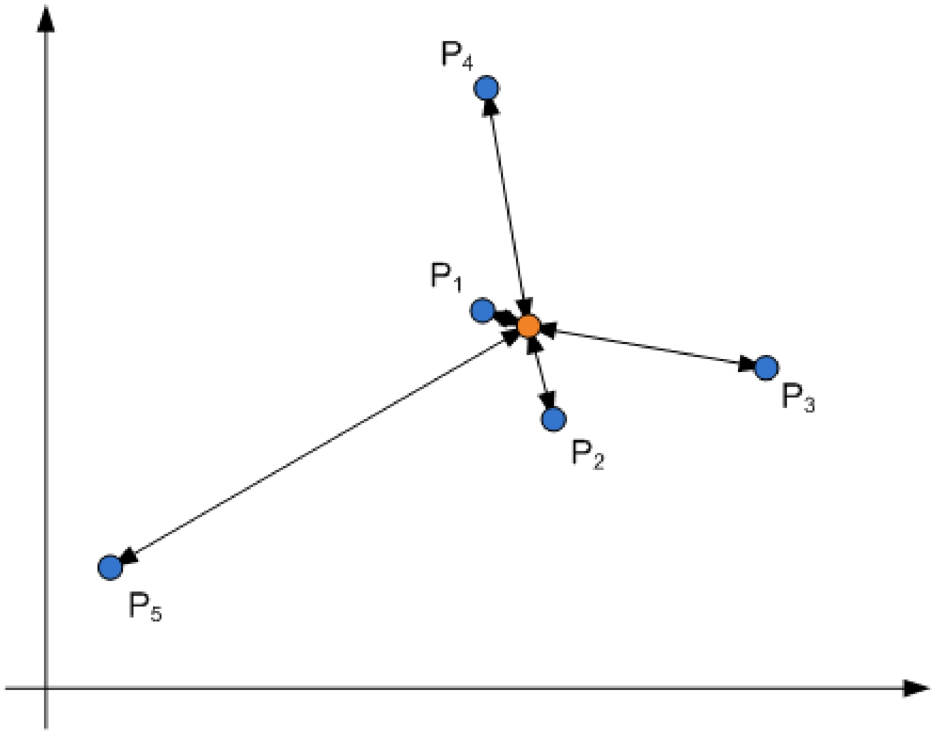

Abbildung 3.4: Nächste Nachbarn eines Approximationspunktes

In der folgenden Tabelle sind die Abstände und absoluten Fehler der Nachbarn dargestellt.

Nachbarpunkt	Abstand zur Approximationsstelle	inverser Abstand	Wichtung	absoluter Fehler
P_1	0.1	10	$\frac{10}{20}$	-0.1
P_2	0.2	5	$\frac{5}{20}$	0.5
P_3	0.5	2	$\frac{2}{20}$	-0.2
P_4	0.5	2	$\frac{2}{20}$	-1
P_5	1	1	$\frac{1}{20}$	2

Tabelle 3.2: Fiktive Werte für die fünf nächsten Nachbarn

Somit ergibt sich die Schätzung des absoluten Fehlers $\Delta(\mathbf{x}^*)$ an der Approximationsstelle $(\mathbf{x}^*)$ aus:

$$\Delta(\mathbf{x}^*) = \frac{10}{20} \cdot (-0.1) + \frac{5}{20} \cdot 0.5 + \frac{2}{20} \cdot (-0.2) + \frac{2}{20} \cdot (-1) + \frac{1}{20} \cdot 2 = \underline{\underline{0.055}}$$

Man erkennt sehr deutlich die Auswirkung der Gewichtung mittels des inversen Abstandes auf das Ergebnis der Schätzung. Hätte man nur das arithmetische Mittel der absoluten Fehler der fünf Nachbarn berechnet, so läge die Schätzung bei einem Wert von 1.2.

4 Vergleich der einzelnen Approximationsverfahren

In diesem Kapitel sollen die vorgestellten Approximationsverfahren für mehrdimensionale Daten anhand einiger analytischer Testfunktionen und technischer Testdaten bewertet werden. Dabei soll zum einen mit Hilfe der Gütekriterien festgestellt werden, welches Verfahren für welche Art von Testdaten besonders geeignet ist, zum anderen die Rechenzeit der Verfahren miteinander verglichen werden.

4.1 Vorgehen

Bei den zu untersuchenden analytischen Testfunktionen wird zunächst der Bereich festgelegt, über dem die Funktion approximiert werden soll. Anschließend werden zufällig jeweils 30 Trainingsmengen mit 100, 1000 und 10000 Punkten über diesem Definitionsbereich generiert. Diese Punkte werden in einer Datei abgespeichert, damit für jedes Verfahren die gleichen Trainingsmengen verwendet werden und somit die Vergleichbarkeit der Ergebnisse gewährleistet ist. Anschließend werden 30 Testmengen mit jeweils 1000 Punkten mithilfe des Zufallszahlengenerators erzeugt. Da bei einigen Verfahren an den Rändern des Definitionsbereiches große Fehler auftreten können, welche die Beurteilung der Güte des Modells stark beeinflussen, werden die untere Grenze für die Zufallszahlengenerierung in alle Dimensionen um 2% vergrößert und die obere Grenze dementsprechend verkleinert.

Anschließend werden für jedes Approximationsverfahren, teilweise zudem mit verschiedenen Einstellungen der Parameter, 30 Modelle unter Verwendung der erzeugten Trainingsmengen aufgebaut und das arithmetische Mittel der Bewertungskriterien berechnet.

Die verwendeten Verfahren sowie Einstellungen sind in Tabelle 4.1 dargelegt, die Bewertungskriterien entsprechen denen aus Tabelle 3.2. Zusätzlich wird das modifizierte Bestimmtheitsmaß, sowohl mit der Gradientenmethode als auch durch Verwendung eines Optimierungsverfahrens zur Bestimmung des minimalen Abstandes, berechnet.

QR-333	
Verfahren	Quasi-Regression
Grad	3
Rang	3
Ordnung	3

QR-555	
Verfahren	Quasi-Regression
Grad	5
Rang	5
Ordnung	5

REG-333	
Verfahren	Regression
Grad	3
Rang	3
Ordnung	3

REG-555	
Verfahren	Regression
Grad	5
Rang	5
Ordnung	5

MLS-333	
Verfahren	Moving Least Squares
Grad	3
Rang	3
Ordnung	3

MLS-555	
Verfahren	Moving Least Squares
Grad	5
Rang	5
Ordnung	5

Sg-2	
Verfahren	globaler Shepard
Glättungsparameter	2

Sg-4	
Verfahren	globaler Shepard
Glättungsparameter	4

Sl-2	
Verfahren	lokaler Shepard
Glättungsparameter	2

KRI	
Verfahren	Kriging

RBF	
Verfahren	Radiale Basisfunktionen
Anzahl der Iterationen des Neural Gas	20
Anzahl der Iterationen für das RBF-Training	50

Tabelle 4.1: Untersuchte Approximationsverfahren und verwendete Einstellungen

4.2 Analytische Testfunktionen

Im Folgenden werden zwei analytische Testfunktionen aus [11] vorgestellt, an denen die Approximationsverfahren untersucht werden. Während die dreidimensionale Griewank-Funktion, eine Funktion mit drei Variablen, eine Funktion mit mehreren Minima und Maxima mit teilweise recht großen Gradienten darstellt, verläuft die Six-hump camel back-Funktion, eine Funktion mit zwei Variablen, deutlich flacher.

Dreidimensionale Griewank-Funktion

Die allgemeine Griewank-Funktion wird beschrieben durch:

$$f\left(\mathbf{x}\right)=\frac{1}{4000}\cdot\sum_{j=1}^{m}{x_j}^2-\prod_{j=1}^{m}\cos\left(\frac{x_j}{\sqrt{j}}\right)+1 \tag{4.1}$$

Dementsprechend lautet der funktionale Zusammenhang der dreidimensionalen Griewank-Funktion:

$$f\left(x_1,x_2,x_3\right)=\frac{{x_1}^2+{x_2}^2+{x_3}^2}{4000}-\cos\left(x_1\right)\cdot\cos\left(\frac{x_2}{\sqrt{2}}\right)\cdot\cos\left(\frac{x_3}{\sqrt{3}}\right)+1 \tag{4.2}$$

Diese Funktion wird im Intervall $[-4,4]^3$ untersucht, d.h. die Trainingspunkte werden in diesem Intervall gewählt, die Testpunkte aus $[-3.84,3.84]^3$. Die Abbildung 4.1 zeigt den Graphen der zweidimensionalen Griewank-Funktion.

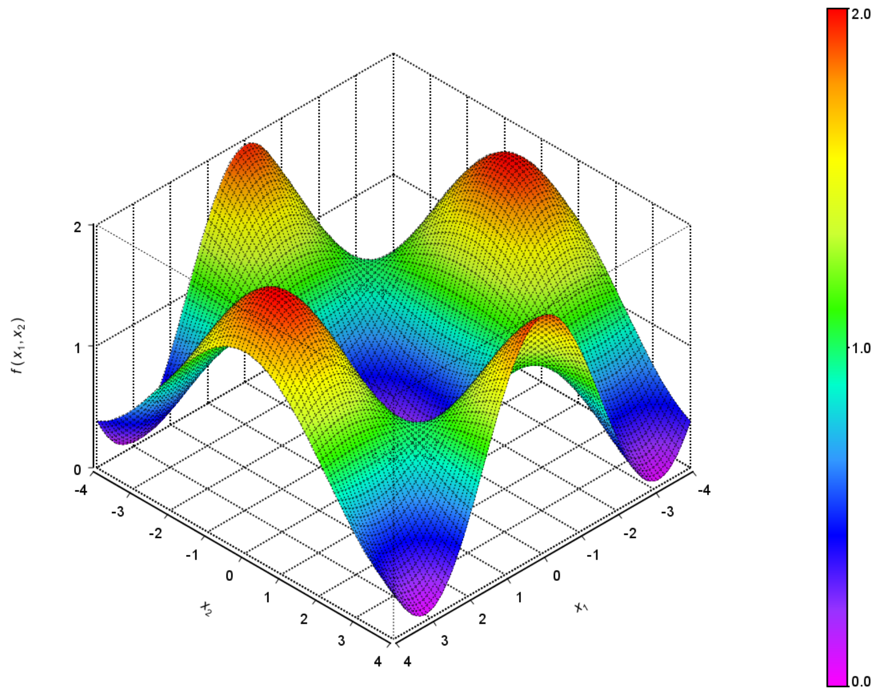

Abbildung 4.1: Griewank-Funktion

Six-hump camel back-Funktion

Die Six-hump camel back-Funktion lässt sich darstellen durch:

$$f(x_1, x_2) = \left(4 - 2.1 \cdot {x_1}^2 + \frac{{x_1}^4}{3}\right) \cdot {x_1}^2 + x_1 \cdot x_2 + \left(-4 + 4 \cdot {x_2}^2\right) \cdot {x_2}^2 \tag{4.3}$$

Diese Funktion wird im Intervall $[-2, 2] \times [-1, 1]$ untersucht und dementsprechend die Testpunkte aus $[-1.92, 1.92] \times [-0.96, 0.96]$ gewählt. Die Abbildung 4.2 visualisiert die Testfunktion, welche von nun an mit CamelBack abgekürzt wird.

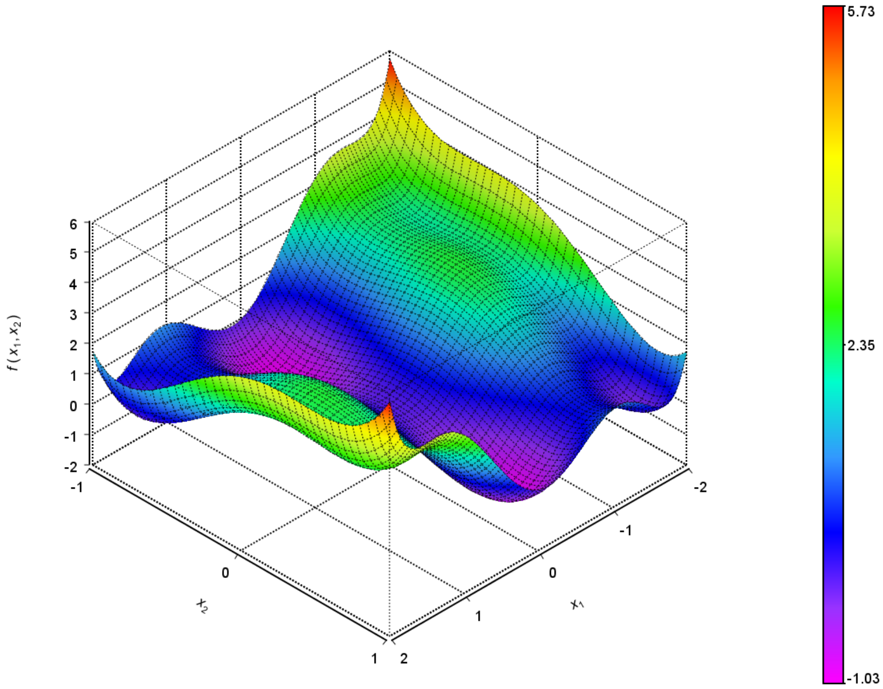

Abbildung 4.2: Six-hump camel back-Funktion

4.3 Technische Anwendung

Im Folgenden werden Daten einer Optimierungsausgabe aus dem Modul V-CD untersucht. (vgl. Anhang A.2) Das Ziel dieser Aufgabe bildet die Geometrieoptimierung eines Kettentriebes. Mit Hilfe von zwölf Parametern, unter anderem der Geometrie der Schienen und des Spannelementes, werden sechs Ziele verfolgt (vgl. [13]):

f_1 : Minimierung des Kolbenhubs des Spannelements,

f_2 : Verhältnis von Vorspannkraft zu Spannerkraft etwa 2.5,

f_3 : Minimierung der Änderung dieses Verhältnisses zwischen gelängtem und ungelängtem Zustand,

f_4 : Länge der freien Trumstücke sollte etwa vier Kettenglieder betragen,

f_5 : Minimierung der Reibleistung und

f_6 : Minimierung der Kontaktlänge über Ein- und Auslaufkreisbögen.

Unter Verwendung des Optimierungstools der IAV GmbH wurden jeweils die Funktionswerte an mehr als 10000 Punkten berechnet, um dann analog zu den analytischen Testfunktionen für 100, 1000 und 10000 Punkte zufällig jeweils 30 verschiedene Trainingsmengen sowie Testmengen zusammengestellt. Im Gegensatz zu den analytischen Funktionen kann jedoch nicht damit gerechnet werden, dass die Trainingsmengen den Datenraum annähernd gleichmäßig abdecken. Stattdessen muss man feststellen, dass sich mehrere Punktwolken bilden und sich in einigen Bereichen des Raumes kein einziger Punkt befindet, was darin begründet ist, dass viele Konstellationen der zwölf Parameter technisch nicht realisierbar sind.

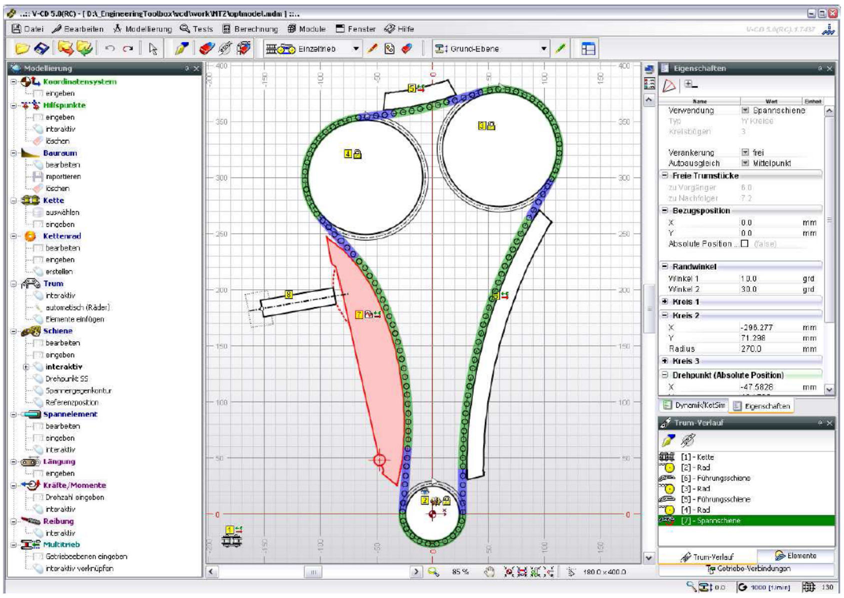

Abbildung 4.3: Modellierung eines Kettentriebes mit der Anwendung V-CD

4.4 Approximationsergebnisse

In den folgenden Tabellen wird die Güte der Approximation bei den verschiedenen Beispielen dargestellt. Dabei wurden die Bewertungskriterien für jeden der 30 Durchläufe ermittelt und der Median der Ergebnisse berechnet. Es wird nicht das arithmetische

Mittel verwendet, da der relative Fehler bei Funktionswerten in der Nähe von null sehr große Werte annehmen kann und dieser das Ergebnis stark verfälschen kann. Bei allen anderen Gütekriterien unterscheiden sich Median und arithmetisches Mittel nur unwesentlich.

Dreidimensionale Griewank-Funktion

Die Tabelle 4.2 verdeutlicht die berechneten Werte der Gütekriterien für die dreidimensionale Griewank-Funktion, welche im Intervall $[-4, 4]^3$ untersucht wurde und dabei Funktionswerte im Intervall $[0, 2]$ annimmt, unter Verwendung von jeweils 100 Trainingspunkte und 1000 Testpunkte. Die Mengen der Testpunkte wurden für diese Arbeit unabhängig von der Anzahl der Trainingspunkte gewählt, um die berechneten Ergebnisse möglichst gut miteinander vergleichen zu können. Um so viele Informationen wie möglich nutzen zu können, wird man in der Praxis möglichst viele Punkte für den Aufbau des Modells verwenden und mit nur wenigen Punkten die Güte des Modells bestimmen.

Verfahren	Δ_{Max}	Δ_{Mean}	ε_{Max}	ε_{Mean}	Δ_{MQ}	b	b_{ModG}	b_{ModO}
QR-333	1.0899	0.2445	21.523	0.3641	0.1085	-0.141	-0.138	-0.124
QR-555	1.3088	0.2496	19.276	0.3567	0.1101	-0.133	-0.038	0.0385
REG-333	1.1622	0.2472	21.227	0.3690	0.1115	-0.184	-0.165	-0.146
REG-555	5.3585	0.3436	11.543	0.4312	0.3456	-2.627	-0.912	-0.217
MLS-333	1.8501	0.1850	16.357	0.2721	0.0716	0.2229	0.2972	0.3830
MLS-555	4.2427	0.2024	8.3413	0.2650	0.1518	-0.623	0.1639	0.4882
Sg-2	0.8985	0.1840	17.314	0.2792	0.0674	0.2841	0.2968	0.3120
Sg-4	0.8402	0.1493	9.0645	0.2003	0.0432	0.5373	0.5578	0.5742
Sl-2	0.8028	0.1631	4.8243	0.2017	0.0480	0.5032	0.5296	0.6662
KRI	0.5895	0.0573	1.2410	0.0653	0.0092	0.9051	0.9073	0.9107
RBF	0.9775	0.2083	18.030	0.3201	0.0830	0.1393	0.1499	0.1499

Tabelle 4.2: Güte der Approximation für Griewank mit 100 Trainingspunkten

Analysiert man die Ergebnisse der Bewertungskriterien, so stellt man schnell fest, dass das Kriging-Verfahren die mit Abstand besten Approximationsergebnisse liefert. Mit

einem Bestimmtheitsmaß von etwa 0.9 kann man davon sprechen, dass die Originalfunktion unter Verwendung von nur 100 Punkten bereits sehr gut abgebildet wurde.
Im Gegensatz dazu schneiden die Verfahren Regression und Quasi-Regression mit recht großen absoluten sowie relativen Fehlern und einem Bestimmtsheitsmaß kleiner null sehr schlecht ab. Selbst die modifizierten Bestimmtheitsmaße nehmen nur unwesentlich größere Werte an, was darauf schließen lässt, dass sich die unbefriedigenden Ergebnisse nicht durch eine Verschiebung der Approximationsfunktion gegenüber der Originalfunktion begründen lässt. Des Weiteren lässt sich feststellen, dass die Quasi-Regression trotz der Verwendung einer Näherung etwas bessere Ergebnisse liefert als die Regression. Die Methode der Moving Least Squares, bei welcher die Approximationswerte mit deutlich mehr Aufwand ermittelt werden, liefert für dieses Beispiel sichtbar bessere Ergebnisse. Zwar sind die maximalen absoluten Fehler sehr groß, jedoch die mittleren Fehler wesentlich kleiner. Vor allem die modifizierten Bestimmtheitsmaße bewerten das Verfahren deutlich besser als Regression und Quasi-Regression.
Vergleicht man für die Verfahren der Approximation im quadratischen Mittel die beiden Verfahrenseinstellungen 333 und 555 miteinander, so lässt sich bemerken, dass obwohl für die Einstellung Maximalrang 5, Maximalgrad 5, Maximalordnung 5 deutlich mehr und höhergradige Ansatzfunktionen verwendet werden, die Ergebnisse kaum besser oder teilweise sogar schlechter sind. Dies lässt vermuten, dass die Approximationsfunktionen in diesen Fällen zum Überschwingen neigen.
Die Verfahren der Shepard-Interpolation liefern in Anbetracht der geringen Anzahl an Trainingspunkten recht gute Ergebnisse. Deutlich zu erkennen ist der Einfluss des Glättungsparameters, welcher jedoch erfahrungsgemäß bei der lokalen Variante des Verfahren deutlich weniger Einfluss auf das Ergebnis der Approximationsfunktion besitzt.
Wenig überraschend erscheint das schlechte Abschneiden der radialen Basisfunktionen. Bei nur 100 Trainingspunkten können nur sehr wenige Gaußglocken als Ansatzfunktionen verwendet werden, mit denen sich der Raum nur schlecht abdecken lässt.
Es lässt sich jedoch vermuten, dass mit zunehmender Anzahl der Trainingspunkte dies Verfahren deutlich bessere Approximationsmodelle liefert, was Tabelle 4.3 auch untermauert.

Werden zum Aufbau des Modells anstatt 100 Punkten die zehnfache Anzahl verwendet, so verbessern sich die Ergebnisse schon sehr deutlich. Für jedes Verfahren nimmt das Bestimmtheitsmaß einen positiven Wert an. Die Verfahren Regression und Quasi-Regression sind nicht nur erneut die mit Abstand schlechtesten, auch die Verbesserung im Vergleich zur Verwendung von 100 Punkten fällt für diese Verfahren am geringsten aus. Es lässt sich jedoch nun feststellen, dass wie erwartet bei Verwendung

Verfahren	Δ_{Max}	Δ_{Mean}	ε_{Max}	ε_{Mean}	Δ_{MQ}	b	b_{ModG}	b_{ModO}
QR-333	1.0076	0.2207	22.521	0.3361	0.0938	0.0070	0.0070	0.0112
QR-555	0.9420	0.1956	19.933	0.2903	0.0709	0.2471	0.2593	0.2829
REG-333	1.0125	0.2212	22.542	0.3359	0.0939	0.0060	0.0061	0.0106
REG-555	1.0627	0.1963	19.466	0.2882	0.0710	0.2404	0.2655	0.2975
MLS-333	0.7456	0.1291	16.723	0.2073	0.0324	0.6569	0.6640	0.6752
MLS-555	0.4390	0.0614	9.5868	0.1024	0.0078	0.9177	—	—
Sg-2	0.8082	0.1677	17.055	0.2570	0.0562	0.4090	0.4494	0.5449
Sg-4	0.4795	0.0698	5.7652	0.1038	0.0103	0.8909	0.9023	0.9127
Sl-2	0.4244	0.0726	2.2750	0.0914	0.0097	0.8966	0.9059	0.9452
KRI	0.0092	0.0004	0.0147	0.0004	7.1e-7	1.0000	1.0000	1.0000
RBF	0.5824	0.0801	2.4511	0.1011	0.0143	0.8509	0.8533	0.8568

Tabelle 4.3: Güte der Approximation für Griewank mit 1000 Trainingspunkten

von höhergradigen und einer größeren Anzahl von Ansatzfunktionen wesentlich bessere Approximationsergebnisse erzielt werden. Vor allem die Ergebnisse der Moving Least Squares untermauern die Sinnhaftigkeit der deutlich aufwändigeren Berechnung der Modellwerte durch Lösen eines Gleichungssystems für jeden Approximationspunkt im Vergleich zur Regression, bei der nur ein Gleichungssystem für alle Punkte gelöst werden muss. Für das Verfahren MLS-555 wurde auf die Ermittlung der modifizierten Bestimmtheitsmaße verzichtet, da dafür das Modell an sehr vielen Stellen ausgewertet werden muss, was wie erwähnt bei diesem Verfahren einen sehr großen Zeitaufwand in Anspruch nimmt.
Auch die Verfahren der Shepard-Interpolation liefern ähnlich gute Werte, wobei jedoch erwähnt werden muss, dass die globale Variante mit Glättungsparameter vier deutlich bessere Ergebnisse liefert als mit Glättungsparameter zwei und ähnlich wie die lokale Variante des Verfahrens.
Das Kriging-Verfahren ist sogar in der Lage die Originalfunktion annähernd perfekt abzubilden.
Betrachtet man die Ergebnisse der Approximation unter Verwendung von 10000 Trainingspunkten aus Tabelle 4.4 mit denen von 1000 Trainingspunkten, so stellt man nur noch leichte Verbesserungen der Qualität fest. Man sollte jedoch festhalten, dass die

Verfahren der Approximation im quadratischen Mittel die geringsten Verbesserungen aufweisen. Auf die Berechnung der modifizierten Bestimmtheitsmaße musste aus Zeitgründen bei einigen Verfahren verzichtet werden. Zudem wurde kein Modell mit Hilfe des Kriging-Verfahrens aufgebaut, da dies viel zu viel Zeit in Anspruch nimmt.

Verfahren	Δ_{Max}	Δ_{Mean}	ε_{Max}	ε_{Mean}	Δ_{MQ}	b	b_{ModG}	b_{ModO}
QR-333	1.0267	0.2251	65.788	0.3574	0.0962	0.0229	0.0229	0.0251
QR-555	1.1401	0.1996	56.643	0.3032	0.0737	0.2469	0.2616	0.2868
REG-333	0.9865	0.2186	21.698	0.3337	0.0923	0.0216	0.0216	0.0232
REG-555	0.9013	0.1901	19.011	0.2795	0.0673	0.2773	0.2886	0.3079
MLS-333	0.7295	0.1253	16.124	0.1995	0.0308	0.6736	—	—
MLS-555	0.4176	0.0573	9.1504	0.0946	0.0068	0.9286	—	—
Sg-2	0.7586	0.1619	16.549	0.2478	0.0522	0.4463	—	—
Sg-4	0.2275	0.0308	2.9823	0.0452	0.0021	0.9780	—	—
Sl-2	0.2020	0.0329	0.6337	0.0399	0.0020	0.9787	—	—
RBF	0.4059	0.0458	2.5983	0.0550	0.0037	0.9626	0.9654	0.9685

Tabelle 4.4: Güte der Approximation für Griewank mit 10000 Trainingspunkten

Bei der Untersuchung und Bewertung der Verfahren sollte man jedoch nicht nur die Approximationsergebnisse betrachten; auch die benötigte Rechenzeit darf nicht außer Acht gelassen werden. Die Tabelle 4.5 verdeutlicht die Rechenzeiten für den Aufbau des Modells (mit „T“ gekennzeichnet) und die Funktionswertberechnung der 1000 Testpunkte (mit „FWB“ gekennzeichnet) für die jeweiligen Verfahren und Einstellungen unter Verwendung eines Intel Core 2 Duo Prozessors mit 2,13 GHz und 2 GB RAM. Dabei lässt sich feststellen, dass die Verfahren Regression und Quasi-Regression eine vernachlässigenswerte Rechenzeit in Anspruch nehmen. Da bei der Shepard-Interpolation kein wirkliches Modell aufgebaut wird, sondern für jeden Approximationspunkt die Trainingspunkte gewichtet werden, wird „nur“ Zeit für die Funktionswertberechnung in Anspruch genommen. Diese fällt für dieses Beispiel noch recht gering aus, jedoch lässt sich festhalten, dass die lokale Variante mehr als die doppelte Zeit in Anspruch nimmt als die globale Variante des Verfahrens.
Bei den radialen Basisfunktionen sowie dem Kriging-Verfahren muss der Großteil der Zeit in den Aufbau des Modells investiert werden, die Funktionswertberechnung erfolgt anschließend vergleichsweise schnell, weshalb auch die Berechnung des modifizierten

Bestimmtheitsmaßes bei diesen Verfahren stets möglich ist. Die Zeit für den Aufbau des Modells steigt jedoch beim Kriging-Verfahren mit der Anzahl der Trainingspunkte sehr schnell an, weshalb dieses Verfahren bei mehr als 1000 Trainingspunkten kaum anwendbar ist.
Im Gegensatz dazu nimmt bei den Moving Least Squares die Funktionswertberechnung die viel größere Zeit in Anspruch, weshalb auch die Berechnung des modifizierten Bestimmtheitsmaßes kaum möglich ist.

TP / Verfahren	100		1000		10000	
	T in s	FWB in s	T in s	FWB in s	T in s	FWB in s
QR-333	<< 1	<< 1	<< 1	<< 1	0.09	<< 1
QR-555	<< 1	<< 1	<< 1	<< 1	0.2	<< 1
REG-333	<< 1	<< 1	<< 1	<< 1	0.3	<< 1
REG-555	<< 1	<< 1	0.2	<< 1	2.2	<< 1
MLS-333	<< 1	0.75	<< 1	7.2	<< 1	73.2
MLS-555	<< 1	8.8	<< 1	54.7	0.1	714
Sg-2/Sg-4	0	<< 1	0	0.4	0	4.0
Sl-2	0	0.2	0	1.4	0	10.6
KRI	4.9	<< 1	2800	6.7	—	—
RBF	0.3	<< 1	8.9	<< 1	317	0.3

Tabelle 4.5: Rechenzeiten für die Approximation der Griewank-Funktion

Six-hump camel back-Funktion

Für die Approximation der Six-hump camel back-Funktion (CamelBack), welche im Intervall

$$[-2, 2] \times [-1, 1]$$

untersucht wurde und dabei Funktionswerte im Intervall $[-1.03, 5.73]$ annimmt, wurden die gleichen Anzahlen an Trainings- und Testpunkten verwendet wie für die Griewank-Funktion. Die Approximationsergebnisse für 100 Trainingspunkte sind in Tabelle 4.6 dargestellt. Dabei lässt sich feststellen, dass alle Verfahren trotz der geringen Anzahl an Trainingspunkten recht gute Ergebnisse liefern und deutlich bessere als bei der Griewank-Funktion. Der Grund dafür ist vor allem darin zu suchen, dass die

CamelBack-Funktion wesentlich geringere Gradienten aufweist.

Verfahren	Δ_{Max}	Δ_{Mean}	ε_{Max}	ε_{Mean}	Δ_{MQ}	b	b_{ModG}	b_{ModO}
QR-333	1.7571	0.5632	593.69	3.2403	0.4484	0.6327	0.8195	0.9205
QR-555	2.3211	0.4715	563.67	2.5929	0.3473	0.7221	0.8887	0.9721
REG-333	1.4690	0.5348	569.21	3.1205	0.3987	0.6754	0.8270	0.9092
REG-555	1.5726	0.3340	412.94	2.2450	0.1606	0.8679	0.9380	0.9796
MLS-333	1.0515	0.3490	498.04	2.3884	0.1662	0.8621	0.9199	0.9531
MLS-555	1.1129	0.1598	226.38	1.1813	0.0392	0.9676	0.9843	0.9936
Sg-2	1.6519	0.3585	482.37	2.6999	0.2106	0.8293	0.9065	0.9834
Sg-4	1.2639	0.2088	256.32	1.1453	0.0810	0.9339	0.9705	0.9916
Sl-2	1.4396	0.2371	248.07	1.2300	0.1077	0.9120	0.9328	0.9922
KRI	0.1937	0.0190	20.701	0.1066	0.0008	0.9993	0.9997	0.9999
RBF	1.4654	0.3189	397.10	2.0994	0.1599	0.8702	0.9279	0.9581

Tabelle 4.6: Güte der Approximation für CamelBack mit 100 Trainingspunkten

Erwartungsgemäß schneidet das Kriging-Verfahren erneut am besten ab. Mit diesem Verfahren lässt sich die CamelBack-Funktion bereits mit 100 Trainingspunkten fast perfekt approximieren. Das Verfahren Moving Least Squares mit Maximalrang 5, Maximalgrad 5, Maximalordnung 5 (MLS-555) liefert die zweitbesten Ergebnisse, während die Resultate bei der Griewank-Funktion mit 100 Punkten sehr schlecht ausfielen. Die Ursache könnte darin liegen, dass bei einer dreidimensionalen Funktion bei den gleichen Einstellungen der Verfahrensparameter deutlich mehr Ansatzfunktionen verwendet werden und dann die Anzahl der Funktionen im Vergleich zu nur 100 Punkten viel zu groß wird. Die Ergebnisse aller anderen Versionen der Approximation im quadratischen Mittel fallen zwar wesentlich schlechter aus, jedoch sind diese im Gegensatz zur Griewank-Funktion durchaus brauchbar, zumal der Vorteil dieser Verfahren in der Rechenzeit zu finden ist. (vgl. Tabelle 4.9 Die Shepard-Interpolation sowohl als globale als auch als lokale bringt sehr gute Resultate. Dabei lässt sich wieder feststellen, dass die Variante Sg-4 mit größerem Glättungsparameter deutlich besser abschneidet, was aufgrund der Struktur der CamelBack-Funktion auch nicht verwundert. Die radialen Basisfunktionen liefern bei 100 Trainingspunkte vergleichsweise noch recht große Approximationsfehler, was sich jedoch bereits bei Verwendung von 1000 Trainingspunkten (vgl. Tabelle 4.7) stark ändert.

Verfahren	Δ_{Max}	Δ_{Mean}	ε_{Max}	ε_{Mean}	Δ_{MQ}	b	b_{ModG}	b_{ModO}
QR-333	1.4592	0.5221	504.69	2.9626	0.3790	0.6931	0.8313	0.9160
QR-555	1.0022	0.3399	376.29	2.2670	0.1533	0.8778	0.9405	0.9769
REG-333	1.3523	0.5210	588.69	3.0126	0.3730	0.6977	0.8390	0.9175
REG-555	0.7464	0.3280	385.78	2.2325	0.1360	0.8898	0.9467	0.9792
MLS-333	0.8161	0.3371	436.49	2.3550	0.1505	0.8783	0.9296	0.9553
MLS-555	0.3197	0.1482	224.21	1.0386	0.0280	0.9772	0.9888	0.9946
Sg-2	0.9802	0.2420	352.15	1.7844	0.0929	0.9259	0.9680	0.9974
Sg-4	0.5167	0.0624	55.467	0.3087	0.0079	0.9935	0.9977	0.9996
Sl-2	0.7117	0.0775	64.571	0.3761	0.0128	0.9896	0.9924	0.9995
KRI	0.0050	0.0004	0.3062	0.0022	3.7e-7	1.0000	1.0000	1.0000
RBF	0.6708	0.1445	156.60	0.6781	0.0374	0.9694	0.9849	0.9932

Tabelle 4.7: Güte der Approximation für CamelBack mit 1000 Trainingspunkten

Beim Vergleich der Approximationsergebnisse mit 100 und 1000 Trainingspunkten fällt zudem sehr schnell auf, dass sich die Güte für die Verfahren Regression und Moving Least Squares nur unwesentlich verbessert, während sich die Ergebnisse von Quasi-Regression und Regression deutlich annähern, was damit begründet werden kann, dass sich der systematische Fehler der Abschätzung der Summen mit wachsender Anzahl an Trainingspunkten immer mehr verringert. Bei Verwendung von 10000 Punkten ist dieser Fehler kaum noch spürbar. (vgl. Tabelle 4.8)
Die durchweg guten bis sehr guten Approximationsergebnisse verdeutlichen auch die grafischen Darstellungen der Resultate in Anhang B.
Betrachtet man die Rechenzeiten der Verfahren in Tabelle 4.9 und vergleicht diese mit denen der Griewank-Funktion in Tabelle 4.5, so lässt sich erkennen, dass bei den Verfahren der Approximation im quadratischen Mittel der größte Unterschied zu finden ist. Bei diesen Verfahren wirkt sich die zusätzliche Dimension der Griewank-Funktion und damit verbunden die höhere Anzahl an Ansatzfunktionen aus. Bei allen anderen Methoden fällt eine zusätzliche Dimension bei der Rechenzeit nicht so stark ins Gewicht.

Verfahren	Δ_{Max}	Δ_{Mean}	ε_{Max}	ε_{Mean}	Δ_{MQ}	b	b_{ModG}	b_{ModO}
QR-333	1.3298	0.5212	587.52	3.0708	0.3723	0.6994	0.8392	0.9162
QR-555	0.6776	0.3291	373.13	2.1127	0.1368	0.8894	0.9455	0.9785
REG-333	1.3127	0.5201	559.18	3.0195	0.3712	0.6997	0.8403	0.9171
REG-555	0.6479	0.3282	365.05	2.1478	0.1352	0.8902	0.9468	0.9788
MLS-333	0.8005	0.3353	493.05	2.3574	0.1491	0.8794	0.9298	0.9549
MLS-555	0.2918	0.1479	219.89	1.0514	0.0278	0.9775	—	—
Sg-2	0.6890	0.1811	267.51	1.4190	0.0512	0.9580	0.9848	0.9987
Sg-4	0.1604	0.0181	16.075	0.0944	0.0006	0.9995	0.9998	1.0000
Sl-2	0.2275	0.0246	17.312	0.1235	0.0013	0.9990	0.9992	1.0000
RBF	0.0883	0.0151	16.026	0.0775	0.0004	0.9997	0.9999	1.0000

Tabelle 4.8: Güte der Approximation für CamelBack mit 10000 Trainingspunkten

Verfahren \ TP	100		1000		10000	
	T in s	FWB in s	T in s	FWB in s	T in s	FWB in s
QR-333	<< 1	<< 1	<< 1	<< 1	<< 1	<< 1
QR-555	<< 1	<< 1	<< 1	<< 1	0.1	<< 1
REG-333	<< 1	<< 1	<< 1	<< 1	0.1	<< 1
REG-555	<< 1	<< 1	<< 1	<< 1	0.4	<< 1
MLS-333	<< 1	0.2	<< 1	2.0	<< 1	19.2
MLS-555	<< 1	0.8	<< 1	7.8	<< 1	80.0
Sg-2/Sg-4	0	<< 1	0	0.3	0	3.7
Sl-2	0	0.1	0	0.6	0	3.8
KRI	4.4	<< 1	2650	6.1	—	—
RBF	0.2	<< 1	6.0	<< 1	250	<< 1

Tabelle 4.9: Rechenzeiten für die Approximation der CamelBack-Funktion

V-CD-Beispiel

Da bei dem Anwendungsbeispiel aus dem Modul V-CD mehrere Funktion approximiert werden, wurden im Gegensatz zu den analytischen Testfunktionen nicht alle globalen Bewertungskriterien berechnet, sondern nur die Bestimmtheitsmaße der sechs Zielfunktionen untersucht. Da bei diesem Kriterium die Funktionswerte normiert werden, lassen sich die Ergebnisse bei mehreren Funktionen deutlich besser vergleichen als mit absoluten, relativen oder quadratischen Fehlern. Auf die Berechnung der modifizierten Bestimmtheitsmaße musste aus Zeitgründen verzichtet verwenden. Die Approximationsergebnisse bei Verwendung von 100 Trainingspunkten, d.h. die mittleren ermittelten Bestimmtheitsmaße für alle sechs Zielfunktionen, sind in Tabelle 4.10 dargestellt. Durch die Untersuchung mehrerer Funktionen sowie der sehr hohen Anzahl an Parametern (zwölf) und der damit verbundenen deutlich höheren Rechenzeit, vgl. Tabelle 4.14, konnten einige Verfahren bzw. Einstellungen der Verfahren nicht untersucht werden. So konnte das Verfahren Moving Least Squares nicht getestet werden, da die Bestimmung der Funktionswerte für 1000 Testpunkte bei sehr großen Matrizen zu viel Zeit in Anspruch nimmt. Zudem wurde auf die Untersuchung von QR-555 und REG-555 verzichtet, da bei diesen Einstellungen 6188 Ansatzfunktionen verwendet würden. Bereits die Verfahren QR-333 und REG-333 weisen aufgrund der hohen Anzahl an Parametern 455 Ansatzfunktionen auf.

Verfahren	$b(f_1)$	$b(f_2)$	$b(f_3)$	$b(f_4)$	$b(f_5)$	$b(f_6)$
QR-333	$<< 0$	$<< 0$	$<< 0$	$<< 0$	$<< 0$	$<< 0$
REG-333	$<< 0$	$<< 0$	$<< 0$	$<< 0$	$<< 0$	$<< 0$
Sg-2	0.0377	0.0316	0.0712	0.0646	0.1022	0.1524
Sg-4	0.0745	0.0376	0.0810	0.1041	0.1170	0.1834
Sl-2	0.0753	0.0295	0.0603	0.1169	0.1103	0.1486
KRI	0.1069	0.0373	0.1258	0.0519	0.1510	0.0025
RBF	-0.735	-1.160	-0.349	-0.317	-0.405	0.0046

Tabelle 4.10: Güte der Approximation für V-CD-Beispiel mit 100 Trainingspunkten

Untersucht man die Ergebnisse der Approximation des Datensatzes aus dem Modul V-CD, so fällt auf, dass alle Bestimmtheitsmaße sehr niedrig ausfallen, d.h. Werte sehr nahe bei null annehmen oder sogar negativ sind. Selbst das Kriging-Verfahren liefert wenig überzeugende Ergebnisse. Beim Vergleichen der Ergebnisse mit den analytischen

Testfunktionen darf jedoch nicht vergessen werden, dass bei dem Beispiel aus V-CD die Datenpunkte den Datenraum nicht gleichmäßig überdecken und zudem nicht darauf geachtet werden konnte, dass die Testpunkte nicht am Rand des Definitionsbereiches liegen, wo zumeist die mit Abstand größten Fehler auftreten.
Die Verfahren Regression und Quasi-Regression versagen bei diesem Beispiel unter Verwendung von 100 Trainingspunkten total, was jedoch damit begründet werden kann, dass in diesem Fall deutlich mehr Ansatzfunktionen als Trainingspunkte verwendet wurden, was jedoch im Allgemeinen umgekehrt sein sollte. Von allen anderen Verfahren schneiden die radialen Basisfunktionen am schlechtesten ab, weil sich auch bei diesem Verfahren die geringe Anzahl von 100 Trainingspunkten am stärksten auf die Approximationsergebnisse auswirkt.
Bei der Beurteilung der Approximationsergebnisse, d.h. der ermittelten Bestimmtheitsmaße sollte man jedoch nicht vergessen, dass ein Testpunkt sehr großen Einfluss auf den Wert des Bestimmtheitsmaßes besitzen kann. Wird beispielsweise nur ein Punkt auf dem Rand des Datenbereiches sehr schlecht approximiert, so kann dieser bereits das Ergebnis verfälschen. Deshalb kann bei dem Aufbau eines Ersatzmodells im Modul MCA (vgl. Anhang A.1) auch ein Modell mehrfach trainiert werden, um dann das beste trainierte Modell weiter zu verwenden. Um zu verdeutlichen, welche Unterschiede sich dabei ergeben können, sind in Tabelle 4.11 die besten erzielten Approximationsergebnisse mit den Verfahren dargestellt.

Verfahren	$b(f_1)$	$b(f_2)$	$b(f_3)$	$b(f_4)$	$b(f_5)$	$b(f_6)$
QR-333	$<< 0$	$<< 0$	$<< 0$	$<< 0$	$<< 0$	$<< 0$
REG-333	$<< 0$	$<< 0$	$<< 0$	$<< 0$	$<< 0$	$<< 0$
Sg-2	0.3811	0.2608	0.4346	0.6211	0.4716	0.7608
Sg-4	0.5414	0.3852	0.5691	0.7886	0.6591	0.9138
Sl-2	0.5576	0.3641	0.5631	0.7907	0.6901	0.9216
KRI	0.8122	0.6352	0.8003	0.9331	0.8868	0.9758
RBF	0.4818	0.3461	0.5627	0.7518	0.6178	0.8598

Tabelle 4.11: Güte der besten Approximation für V-CD-Beispiel mit 100 Trainingspunkten

Man erkennt sehr deutlich, wie stark sich die mittleren und besten erzielten Ergebnisse unterscheiden. Bei den besten Ergebnissen überzeugt das Kriging-Verfahren erneut

am meisten. Es lässt sich jedoch feststellen, dass einige Zielfunktionen deutlich besser approximiert werden. So werden bei allen Verfahren die besten Ergebnisse für die Funktionen f_6 und f_5 erzielt, während die Funktion f_2 mit Abstand am schlechtesten angenähert wird. Die Verfahren Regression und Quasi-Regression liefern für das Beispiel aus V-CD bei 100 Trainingspunkten nie brauchbare Ergebnisse.
Bei den analytischen Testfunktionen wurden mit 1000 Trainingspunkten wie zu erwarten jeweils deutlich bessere Approximationsergebnisse erzielt als mit 100. Die Tabelle 4.12 zeigt, dass dies für den Datensatz aus V-CD nur für die radialen Basisfunktionen der Fall ist. Bei allen anderen Verfahren sind die Ergebnisse entweder erneut unbrauchbar (Regression und Quasi-Regression) oder sogar schlechter als bei der Approximation mit 1000 Punkten (Shepard und Kriging). Die Ursache könnte darin liegen, dass bei Verwendung von 1000 Trainingspunkten die Punktwolken deutlich stärker ausgeprägt sind und den Verfahren damit Probleme bereitet werden.

Verfahren	$b(f_1)$	$b(f_2)$	$b(f_3)$	$b(f_4)$	$b(f_5)$	$b(f_6)$
QR-333	$<< 0$	$<< 0$	$<< 0$	$<< 0$	$<< 0$	$<< 0$
REG-333	$<< 0$	$<< 0$	$<< 0$	$<< 0$	$<< 0$	$<< 0$
Sg-2	-0.001	-0.023	0.0446	0.0040	0.0369	0.0666
Sg-4	-0.174	-0.212	-0.100	-0.174	-0.139	-0.080
Sl-2	-0.385	-0.460	-0.376	-0.475	-0.343	-0.370
KRI	-5.638	-5.199	-6.689	-12.28	-7.246	-15.65
RBF	-0.002	-0.056	0.0286	0.0202	0.0763	0.1174

Tabelle 4.12: Güte der Approximation für V-CD-Beispiel mit 1000 Trainingspunkten

Die Tabelle 4.13 jedoch verdeutlicht, dass die besten erzielten Bestimmtheitsmaße bei allen Verfahren deutlich höher sind als bei 100 Punkten. Während das Kriging-Verfahren wiederholt das beste Resultat liefert, rangiert die Regression an zweiter Stelle, obwohl im Mittel mit diesem Verfahren sehr schlechte Approximationsergebnisse erzielt werden. Die Ergebnisse der radialen Basisfunktionen haben sich wie bei allen untersuchten Funktionen bei Erhöhung der Anzahl der Trainingspunkte stark verbessert. Unter den Varianten der Shepard-Interpolation schneidet erneut die globale Version mit dem Glättungsparameter zwei deutlich am schlechtesten ab.

Auf die Darlegung der Approximationsergebnisse bei Verwendung von 10000 Trainings-

Verfahren	$b(f_1)$	$b(f_2)$	$b(f_3)$	$b(f_4)$	$b(f_5)$	$b(f_6)$
QR-333	-11.25	-92.48	-131.5	-6.818	-8.471	-2.355
REG-333	0.9906	0.8207	0.9313	0.9830	0.9969	0.9790
Sg-2	0.4591	0.3715	0.5766	0.6407	0.5556	0.8022
Sg-4	0.7100	0.5944	0.7311	0.8664	0.7915	0.9371
Sl-2	0.7534	0.5889	0.7119	0.8947	0.8294	0.9570
KRI	0.9664	0.9084	0.9587	0.9937	0.9852	0.9961
RBF	0.8750	0.6193	0.7869	0.9298	0.9031	0.9653

Tabelle 4.13: Güte der besten Approximation für V-CD-Beispiel mit 1000 Trainingspunkten

punkte wird verzichtet, da die berechneten Bestimmtsheitsmaße trotz des viel größeren Rechenaufwandes im Mittel keine höheren Werte annahmen als mit 1000 Punkten und bei Betrachtung der besten Approximation die Verfahren sogar wesentlich schlechtere Resultate lieferten.
Die Rechenzeiten für das V-CD-Beispiel, welche in Tabelle 4.14 dargelegt sind, verdeutlichen den enormen Aufwand bei einem Datensatz mit zwölf Parametern und sechs Funktionen.

Verfahren \ TP	100		1000		10000	
	T in s	FWB in s	T in s	FWB in s	T in s	FWB in s
QR-333	2.4	1.5	3.7	1.5	16.5	1.5
REG-333	21.1	1.5	71.1	1.5	865	1.5
Sg-2/Sg-4	0	0.59	0	5.75	0	52.5
Sl-2	0	2.88	0	16.6	0	89.6
KRI	54.4	1.68	16800	37.4	—	—
RBF	21.8	0.17	578	0.4	19700	3.3

Tabelle 4.14: Rechenzeiten für das V-CD-Beispiel

Während sich bei der Shepard-Interpolation die Zeitunterschiede zu den beiden analytischen Testfunktionen mit nur zwei bzw. drei Parametern stark in Grenzen halten und weitestgehend durch die Anzahl der Ziele erklären lassen, wirkt sich die Erhöhung der Anzahl der Parameter auf die anderen Verfahren wesentlich mehr aus. Vor allem bei den Verfahren der Approximation im quadratischen Mittel ist aufgrund der deutlich höheren Anzahl an Ansatzfunktionen ein enormer Zeitanstieg zu erkennen, weshalb auch das Verfahren Moving Least Squares nicht untersucht werden konnte.

Lokale Güte der Approximation

Nachdem die verschiedenen Approximationsverfahren hinsichtlich ihrer Güte betrachtet wurden, soll im Folgenden die Methode der k-nächsten Nachbarn zur Schätzung des absoluten Fehlers an einer Stelle $\mathbf{x}^*$ untersucht werden. Anhand ausgewählter Beispiele soll eine geeignete Anzahl an zu verwendenden Nachbarn ermittelt werden.
Für eine hohe Anzahl an Testpunkten wird die Schätzung im Allgemeinen unabhängig von der Anzahl der Nachbarn recht gute Ergebnisse liefern. In der Praxis jedoch wird man versuchen, von der begrenzten Anzahl an Datenpunkten möglichst viele Informationen in den Aufbau des Approximationsmodells zu stecken und mit nur wenigen Testpunkte die globale Güte des Modells zu schätzen. Deshalb werden zur Bewertung der lokalen Güte nur 30 Testpunkte verwendet. Die Anzahl der für die Schätzung verwendeten Nachbarn wird auf fünf bzw. zehn gesetzt.
Die Abbildung 4.4 zeigt die berechneten, wahren absoluten Fehler einer mit dem Verfahren Quasi-Regression ermittelten Approximationsfunktion der CamelBack-Funktion.

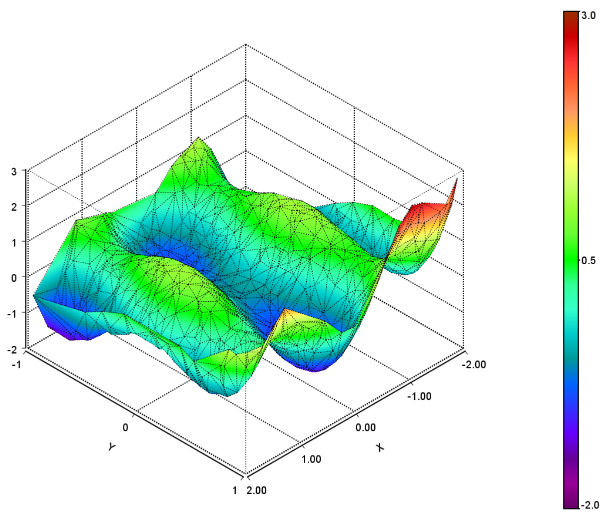

Abbildung 4.4: Absolute Fehler der Quasi-Regression für die CamelBack-Funktion

Damit die Schätzung des absoluten Fehlers besser bewertet werden kann, wurde ein Verfahren gewählt, bei welchem etwas größere absolute Fehler auftreten. In Abbildung 4.5 sind die geschätzten absoluten Fehler der Approximationsfunktion mit der Methode der nächsten Nachbarn unter Verwendung von fünf Nachbarn dargestellt. Vergleicht man die geschätzten Fehler mit den wahren, bei praktischen Beispielen jedoch unbekannten, absoluten Fehlern, so lässt sich feststellen, dass die beiden Darstellungen deutliche Unterschiede aufweisen, die Tendenz der wahren absoluten Fehler bei der Schätzung jedoch durchaus erkannt wurde.

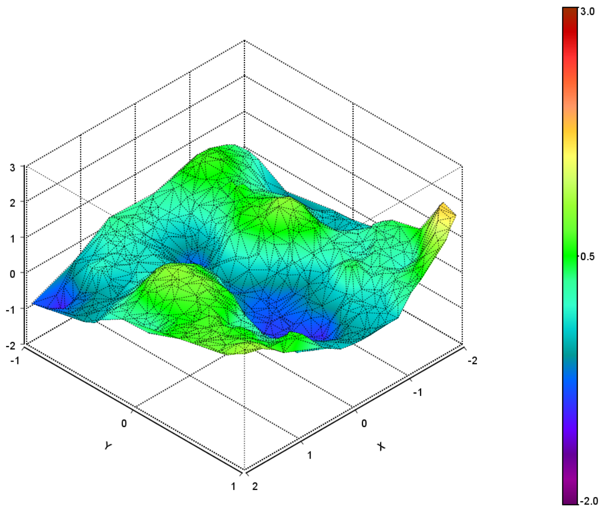

Abbildung 4.5: Schätzung der absoluten Fehler mit den fünf nächsten Nachbarn

Verwendet man für diese Approximationsfunktion wie in Abbildung 4.6 die zehn nächsten Nachbarn, so ändern sich die geschätzten absoluten Fehler nur unwesentlich. Vergleicht man die beiden Schätzungen mit den wahren absoluten Fehlern, so lässt sich erkennen, dass die Schätzung mit den fünf nächsten Nachbarn die etwas besseren Ergebnisse liefert.

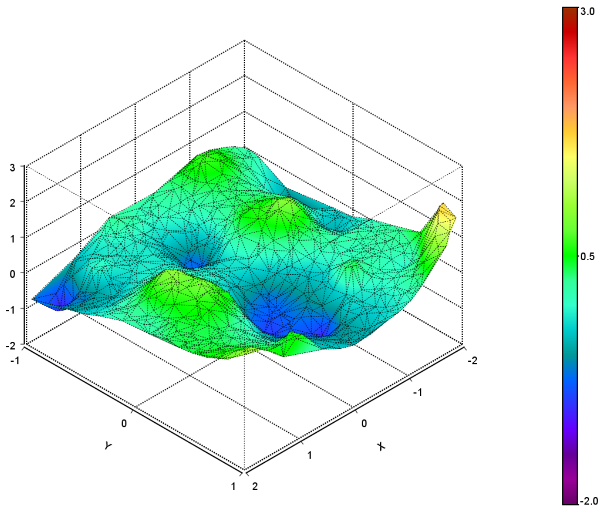

Abbildung 4.6: Schätzung der absoluten Fehler mit den zehn nächsten Nachbarn

Neben der Quasi-Regression wurde noch eine Approximationsfunktion, die mit der Shepard-Interpolation generiert wurde, untersucht. Da für dieses Beispiel der Glättungsparameter der Shepard-Interpolation nicht sehr groß gewählt wurde, treten bei der Darstellung der wahren absoluten Fehler in Abbildung 4.7 einige „Spitzen“ auf.

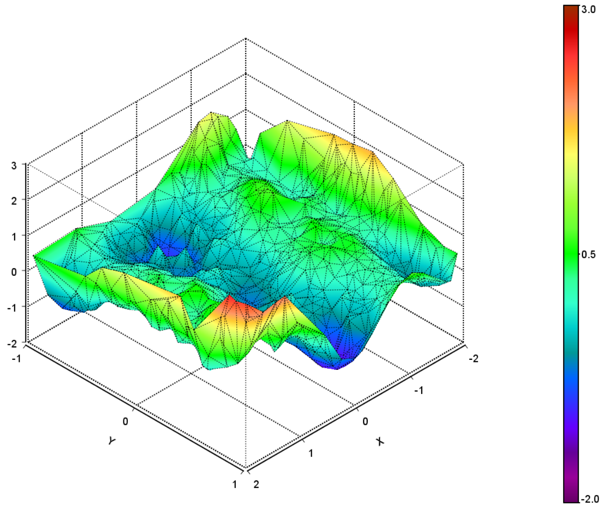

Abbildung 4.7: Absolute Fehler der Shepard-Interpolation für die CamelBack-Funktion

Die in Abbildung 4.8 dargestellte Schätzung der absoluten Fehler mit den fünf nächsten Nachbarn verläuft deutlich flacher als die Funktion der wahren Fehler. Dieses Phänomen tritt bei der Shepard-Interpolation am stärksten auf, da die Approximationsfunktion bei kleinem Glättungsparameter stets große Gradienten aufweist. Diese übertragen sich zumeist auf die Fehlerfunktion und können bei Verwendung von nur 30 Testpunkten nicht annähernd nachgebildet werden.

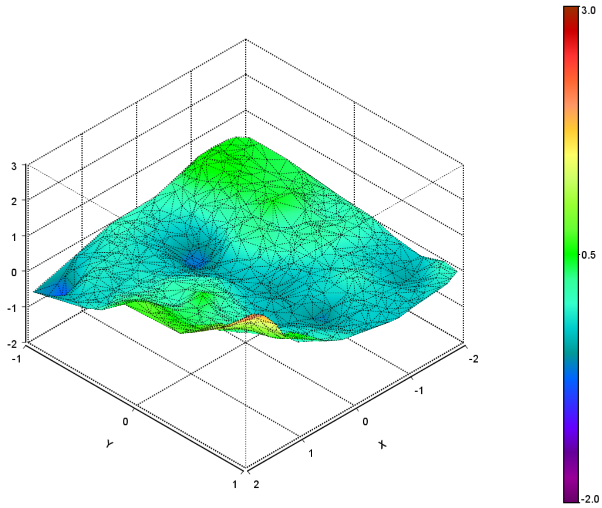

Abbildung 4.8: Schätzung der absoluten Fehler mit den fünf nächsten Nachbarn

Ähnlich wie bei dem mit dem Verfahren Quasi-Regression durchgeführten Beispiel unterscheiden sich die Schätzung mit fünf bzw. zehn Nachbarn in Abbildung 4.9 kaum. Diesmal lässt sich jedoch nicht darüber urteilen, welche der beiden Varianten das bessere Ergebnis liefert.

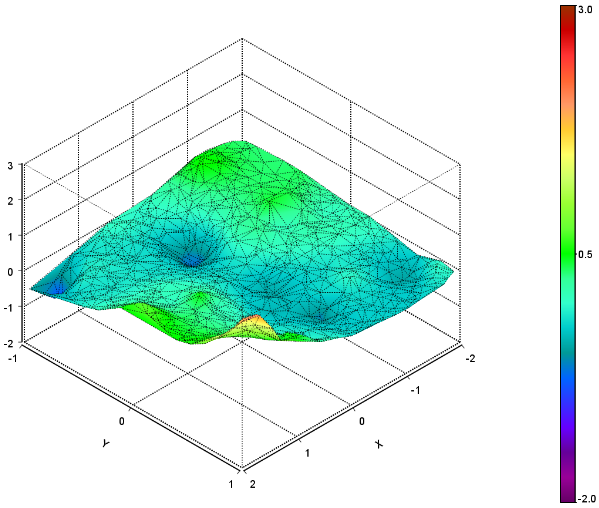

Abbildung 4.9: Schätzung der absoluten Fehler mit den zehn nächsten Nachbarn

Da an den Rändern des Datenbereiches die deutlich größten Fehler auftreten, werden diese bei Verwendung der Nachbarn, die betragsmäßig wesentlich kleinere absolute Fehler aufweisen, stark unterschätzt.

4.5 Auswertung und Empfehlung

Die analytischen Testfunktionen haben gezeigt, dass alle untersuchten Verfahren in der Lage sind, mehrdimensionale Funktionen gut zu approximieren. Vor allem bei einer nicht sehr dynamischen Funktion wie der Six-hump camel back-Funktion werden sehr gute Ergebnisse erzielt. Wie erwartet verbesserten sich bei den analystischen Testfunktionen die Resultate mit zunehmender Anzahl an Trainingspunkten, d.h. je mehr Informationen in das Modell gesteckt werden, desto besser ist das Ergebnis.
Bei beiden Testfunktionen lieferte das Kriging-Verfahren, sofern ein Modell mit diesem Verfahren aufgebaut werden konnte, unabhängig von der Anzahl der Trainingspunkte die besten Resultate. Das Hauptproblem dieses Verfahrens befindet sich jedoch in der Rechenzeit. Aufgrund des aufwändigen Trainings eignet sich das Verfahren für mehr als 500 Trainingspunkte kaum. Zudem sollte auch die Anzahl der Zielfunktionen gering bleiben, um diese Methode der Approximation verwenden zu können. Stehen nur sehr wenige Datenpunkte zur Verfügung, so sollte auf jeden Fall auf dieses Verfahren zurückgegriffen werden, da in diesem Fall die anderen Verfahren zumeist deutlich

schlechtere Ergebnisse liefern.
Die radialen Basisfunktionen bilden das Approximationsverfahren mit dem zeitaufwändigsten Training. Anders als bei dem Kriging-Verfahren sollte man die radialen Basisfunktionen jedoch nicht bei einer sehr kleinen Anzahl von Trainingspunkten verwenden, da dann die Anzahl an verwendeten Gaußglocken für eine ausreichend gute Approximation zumeist zu klein ist. Mit zunehmender Anzahl an Datenpunkten nimmt die Approximationsgüte jedoch recht schnell zu, weshalb dieses Verfahren ab 500 bis 1000 Punkten sehr gut anwendbar ist. Bei einer noch größeren Anzahl hängt die Nutzbarkeit des Verfahrens stark davon ab, wie viel Zeit für die Approximation zur Verfügung steht. Die Rechenzeit wächst jedoch bei weitem nicht so schnell an wie beim Kriging-Verfahren. Zudem lässt sich diese über die Anzahl der Iterationen des Clusteralgorithmus bzw. RBF-Trainings steuern.
Die Verfahren Regression und Quasi-Regression eignen sich besonders für Funktionen, die nicht so große Gradienten aufweisen, da dann die Funktion mit einer geringeren Anzahl von Ansatzfunktionen niederen Grades recht gut nachgebildet werden kann. Dabei sollte bei einer kleinen Anzahl an Trainingspunkten auf jeden Fall die gewöhnliche Approximation im quadratischen Mittel, d.h. die Regression, dem Verfahren Quasi-Regression vorgezogen werden. Bei diesem Szenario wirkt sich der systematische Fehler durch die Abschätzung der Summen und damit verbundenen Vermeidung des Lösens eines Gleichungssystems oft zu stark aus. Mit wachsender Anzahl an Trainingspunkten nähern sich die Ergebnisse mit den Verfahren Quasi-Regression und Regression zumeist an, sodass für eine sehr große Datenmenge das Verfahren Quasi-Regression durchaus seine Berechtigung erzielt. Dies gilt vor allem, wenn die Parameter Grad, Rang und Ordnung gering gewählt werden. Der größte Vorteil der beiden Methoden gegenüber dem Kriging-Verfahren und den radialen Basisfunktionen liegt in der Rechenzeit, die für den Aufbau des Modells benötigt wird. Bei einer geringen Anzahl an Parametern sollte man aufgrund der verschwindend kleinen Rechenzeit auf jeden Fall darüber nachdenken, zuerst Modelle mit den Verfahren Regression oder Quasi-Regression zu erstellen. Erfüllen die aufgebauten Modelle die Güteanforderungen nicht, so sollten die aufwändigeren Verfahren verwendet werden. Der Nachteil der mehrdimensionalen Approximation im quadratischen Mittel ist vor allem darin zu finden, dass dynamische Funktionen stark geglättet werden und bei Verwendung höhergradiger Ansatzfunktionen deren Anzahl mit wachsender Parameteranzahl schnell zunimmt. Dies führt zum einen zur Entstehung großer Matrizen, zum anderen sollte darauf geachtet werden, dass die Anzahl der verwendeten Ansatzfunktionen die Zahl der verwendeten Trainingspunkte nicht übersteigt.
Die bislang beurteilten Verfahren eignen sich besonders, wenn einmal ein Modell aufgebaut werden soll, um dann für sehr viele Punkte die Funktionswerte geschätzt werden

sollen, da der Aufbau des Modells deutlich mehr Zeit in Anspruch nimmt als die Berechnung der Funktionswerte. Bei den Verfahren Moving Least Squares und Shepard-Interpolation ist dies nicht der Fall. Während bei der Shepard-Interpoaltion kein Training des Modells notwendig ist, können bei den Moving Least Squares nur die Werte der Ansatzfunktionen an den Trainingspunkten bestimmt werden. Anschließend muss für jeden zu approximierenden Datenpunkt ein Gleichungssystem gelöst werden.
Das Verfahren Moving Least Squares liefert bei gleicher Wahl der Parameter Grad, Rang und Ordnung in der Regel deutlich bessere Ergebnisse als die Regression. Vor allem bei Verwendung niedergradiger Ansätze kann durch die ortsabhängigen Koeffizienten die Originalfunktion besser nachgebildet werden. Das Hauptproblem dieses Verfahrens liegt jedoch in der Funktionswertberechnung. Sollen für viele Datenpunkte die Funktionswerte geschätzt werden, so nimmt dies sehr viel Zeit in Anspruch, weshalb dieses Verfahren wenig geeignet ist, wenn z. B. auf Knopfdruck Diagramme mit approximierten Punkten erstellt werden sollen. Werden jedoch nur sehr wenige Schätzungen für Funktionswerte benötigt, so kann auf dieses Verfahren zurückgegriffen werden; vor allem bei einer geringen Anzahl an Datenpunkten sollte es dann den anderen Verfahren der Approximation im quadratischen Mittel vorgezogen werden.
Die Shepard-Interpolation bildet wohl das einfachste untersuchte Verfahren. Trotzdem liefert es oft recht stabile Ergebnisse, besonders wenn der Glättungsparameter passend gewählt wird. Dieser hat vor allem bei der globalen Variante dieses Verfahrens einen sehr großen Einfluss auf die Güte der Approximation. Die Ergebnisse der untersuchten Testfunktionen erwecken den Anschein, dass bei einer großen Wahl des Glättungsparamters wesentlich bessere Resultate erzielt werden; es lassen sich jedoch auch Fälle konstruieren, bei denen dies ganz und gar nicht der Fall ist. Die lokale Variante lieferte zumeist ähnlich gute Ergebnisse wie die globale Version mit dem besseren Glättungsparameter. Während die Rechenzeit der lokalen Shepard-Interpolation etwas höher ist, wirkt sich jedoch der Glättungsparameter aufgrund der anders gewählten Funktion für die Shepard-Gewichte deutlich weniger aus, was diese Variante bei unbekannten Daten besser anwendbar macht.
Das praktische Beispiel hat gezeigt, dass man bei komplizierten Problemstellungen, bei denen die Datenpunkte alles andere als gleichmäßig verteilt sind, mehrere Modelle trainieren sollte, da die Aufteilung in Trainings- und Testmenge sehr starken Einfluss auf die Ergebnisse hat. Steht nicht genügend Zeit zur Verfügung, um die Modelle mehrfach zu trainieren, so liefert die Shepard-Interpolation zumeist die stabilsten Ergebnisse, während bei den anderen Verfahren die Schwankungen oft recht groß sind. Zudem verdeutlicht dieses Beispiel, dass die Approximationsgüte keinesfalls für jedes Problem mit der Anzahl der Trainingspunkte wächst.

5 Zusammenfassung und Ausblick

Im Rahmen dieser Diplomarbeit wurden verschiedene Approximationsverfahren für mehrdimensionale Daten vorgestellt. Neben den in der IAV bereits verwendeten Verfahren Kriging und Shepard wurde zum einen die für eindimensionale Daten sehr gebräuchliche Methode der Approximation im quadratischen Mittel aufs Mehrdimensionale angewendet und Spezifikationen dieser Methode analysiert, zum anderen mit den radialen Basisfunktionen ein weiteres speziell für höherdimensionale Probleme anwendbares Verfahren untersucht. Im zweiten Teil der Arbeit wurde nach Kriterien gesucht, anhand denen mathematische Ersatzmodelle, die mit den vorgestellten Approximationsverfahren aufgebaut werden, hinsichtlich ihrer Modellgüte bewertet werden können. Die Eignung der vorgestellten Approximationsverfahren wurden an analytischen Testfunktionen sowie einer technischen Anwendung mithilfe der Gütekriterien nachgewiesen, sowie Empfehlungen für die Nutzung dieser abgeleitet.

Die untersuchten Approximationsverfahren sind in der Lage, analytische Funktionen gut nachzubilden. Bei praktischen Beispielen müssen jedoch in der Regel mehrere Modelle aufgebaut werden, um dann das anhand der Gütekriterien beste Modell verwenden zu können. Liegen die Datenpunkte annähernd gleichverteilt vor, d.h. wird der Datenraum recht gleichmäßig mit Trainingspunkten überdeckt, so liefern die Approximationsverfahren zumeist sehr gute Resultate; die größten Schwierigkeiten treten bei Bildung stark ausgeprägter Punktewolken auf.
Anhand der durchgeführten Gütetests lässt sich für den Einsatz der untersuchten Approximationsmethoden folgende Empfehlung ableiten: Im Mittel lieferte das Kriging-Verfahren, welches den mit Abstand größten Rechenaufwand aufweist, auch die besten Approximationsergebnisse. Bleibt die Anzahl der Datenpunkte in einem kleinen Rahmen, d.h. liegen deutlich weniger als 500 Punkte vor, so kann man damit rechnen, mit diesem Verfahren die besten Ergebnisse in einer noch akzeptablen Rechenzeit zu erzielen. Bei einer größeren Datenmenge könnte man darüber nachdenken weniger Punkte für das Training zu verwenden und die Restmenge als Testmenge für die Beurteilung der Approximationsgüte zu nutzen, um über die geringere Größe des Modells den Rechenaufwand in einem vernünftigen Rahmen zu halten. Stattdessen sollte man jedoch auf ein anderes Verfahren zurückgreifen. Dafür eignen sich besonders die radialen Basis-

funktionen, die mit wachsender Größe der Datenmenge zumeist bessere Modelle liefern. Bei einer sehr großen Anzahl an Trainingspunkten wächst jedoch auch die Rechenzeit für dieses Verfahren spürbar an. Sobald für eine größere Datenmenge in relativer kurzer Zeit Ersatzmodelle erstellt werden sollen, bilden die Verfahren Shepard-Interpolation und Approximation im quadratischen Mittel gute Alternativen. Bei letzterer hängt die Modellgüte stark von den verwendeten Einstellungen sowie der unbekannten Verteilung der Daten ab. Die Verfahren Regression und Quasi-Regression eignen sich besonders, wenn man z. B. eine annähernd lineare oder quadratische Abhängigkeit der Zielfunktionen von den Parametern erwartet. Die Methode Moving Least Squares liefert zwar zumeist noch bessere Ergebnisse, besitzt jedoch den Nachteil, dass bei der Nutzung des Modells zur Schätzung der unbekannten Funktionswerte für jeden neuen Punkt ein Gleichungssystem gelöst werden muss und damit recht viel Zeit in Anspruch genommen wird.

Während bislang im Modul MCA (vgl. Anhang A.1), in welchem aktuell die Hauptanwendung der Approximationsverfahren zur Erstellung mathematischer Ersatzmodelle zu finden ist, beim Aufbau eines Modells für jede Zielfunktion das gleiche Verfahren mit den gleichen Einstellungen verwendet werden muss, sollte in Zukunft darüber nachgedacht werden, den Einsatz verschiedener Methoden für die einzelnene Funktionen zu gewährleisten. Dadurch kann zunächst wie gewohnt mit einem nicht sehr rechenintensiven Verfahren ein Modell erstellt werden und anschließend nur noch die Zielfunktionen mit einer anderen Methode approximiert werden, bei denen das ursprünglich verwendete Verfahren nicht die gewünschten Resultate erzielte.
Auch bei den einzelnen Approximationsmethoden können zum Teil noch Spezifikationen untersucht werden. So lässt sich beim Verfahren Moving Least Squares darüber nachdenken, ähnlich wie bei der lokalen Shepard-Interpolation eine Gewichtsfunktion zu verwenden, die nur die Punkte in einem bestimmten Radius um den Approximationspunkt nutzt. Dieser Ansatz wurde während der Implementierung des Verfahrens bereits umgesetzt, jedoch aufgrund des nicht vorhandenen Zeitgewinns wieder fallen gelassen. Findet man jedoch eine Möglichkeit in kurzer Zeit einen geeigneten Radius zu bestimmen und die Matrizen zu reduzieren, so könnte eine Verringerung des Rechenaufwandes erzielt werden. Bei der globalen Shepard-Interpolation wiederum wurde darüber nachgedacht, mehrere Shepard-Funktionen mit unterschiedlichen Glättungsparametern zu überlagen. Um die optimalen Gewichte der einzelnen Shepard-Funktionen zu bestimmen, müsste man die Datenmenge nicht nur in Trainings- und Testpunkte unterteilen, sondern würde eine dritte Teilmenge benötigen. Die Idee klingt recht vielversprechend, lässt sich jedoch mit der aktuellen Struktur der Approximationsbibliothek recht schwer umsetzen.

Literaturverzeichnis

[1] An, J. und Owen A., B.: *Quasi-Regression*, Journal of Complexity 17, 2001, 588-607.

[2] Arnonkijpanich, B.; Hammer, B.; Hasenfuss, A.: *Local Matrix Adaptation in Topographic Neural Maps*, IfI Technical Report Series, 08/2007, TU Clausthal 2007.

[3] Biehl, M.; et al.: *Stationarity of Matrix Relevance Learning Vector Quantization*, Machine Learning Reports, 01/2009, Universität Leipzig 2009.

[4] Engeln-Müllges, G.; Niederdrenk, K. und Wodicka, R.: *Numerik-Algorithmen*, 9. Auflage, Springer-Verlag, Berlin Heidelberg 2005.

[5] Kammermeier, S.: *Seminar Neuronale Netze und Maschinelles Lernen*, Universität Regensburg, Regensburg 2006.

[6] Kunle, M.: *Entwicklung und Untersuchung von Moving Least Square Verfahren zur numerischen Simulation hydrodynamischer Gleichungen*, Dissertation, Eberhard-Karls-Universität Tübingen, Tübingen 2001.

[7] Kürz, C.: *Radiale-Basis-Funktionen*, Universität zu Köln, Mathematisches Institut, Köln 2008.

[8] Levin, D.: *The Approximation Power of Moving Least-Squares*, Mathematics of Computation, 1998, 1517-1531.

[9] Meisrimel, M.: *Klassische orthogonale Polynome*, Bachelorarbeit, Universität Kassel, Kassel 2008.

[10] Meyer, M.: *Entwicklung eines Expertensystems zur Unterstützung der Auswahl und Parametrisierung von multikriteriellen Optimierungsverfahren für Aufgabenstellungen in der Motorenentwicklung*, Diplomarbeit, TU Chemnitz, Fakultät für Wirtschaftswissenschaften, Chemnitz 2009.

[11] Molga, M.; Smutnicki, C.: *Test functions for optimization needs*, 2005.

[12] Press, W. H.; et al.: *Numerical Recipes: The Art of Scientific Computing*, 3. Auflage, Cambridge University Press, Cambridge 2007.

[13] Röber, M.: *Multikriterielle Optimierungsverfahren für rechenzeitintensive technische Aufgabenstellungen*, Diplomarbeit, TU Chemnitz, Fakultät für Mathematik, Chemnitz 2010.

[14] Schmidt, H.: *Parallelisierung Ersatzmodell-gestützter Optimierungsverfahren*, Diplomarbeit, TU Chemnitz, Fakultät für Mathematik, Chemnitz 2009.

[15] Stöcker, M.: *Untersuchung von Optimierungsverfahren für rechenzeitaufwändige technische Anwendungen in der Motorenentwicklung*, Diplomarbeit, TU Chemnitz, Fakultät für Mathematik, Chemnitz 2007.

[16] Törnig, W.; Spellucci, P.: *Numerische Mathematik für Ingenieure und Physiker, Band 2 Numerische Methoden der Analysis*, 2. Auflage, Springer-Verlag, Berlin Heidelberg 1990.

[17] Yang, G.; Wang, Z.; Deng, W.: *Unbiased generalized quasi-regression*, Computational Statistics and Data Analysis, 2009.

[18] Will, J.; Most, T.: *Metamodell of optimized Prognosis (MoP) - an Automatic Approach for User Friendly Parameter Optimization*, ANSYS Conference & 27^{th} CADFEM Users' Meeting, Leipzig 2009.

Anhang

A EngineeringToolbox

Bei der EngineeringToolbox handelt es sich um eine Sammlung von Softwareprodukten der IAV. Diese Werkzeuge werden den Benutzern unter einer einheitlichen Oberfläche zur Verfügung gestellt. Zudem werden durch dieses Produkt die einzelnen Softwarekomponenten auf dem lokalen Rechner installiert und Software-Updates zur Verfügung gestellt. Bei den Programmen in der EngineeringToolbox handelt es sich zum Teil um externene Produkte, sie beinhaltet jedoch auch viele Eigenentwicklungen. Im Folgenden werden einige Softwareprodukte der EngineeringToolbox, die während der Erstellung dieser Arbeit verwendet wurden, kurz vorgestellt.

A.1 Modul MCA

Das Modul MCA (Multi Criteria Analyzer) wird verwendet, um Ergebnisse einer Optimierung mit Hilfe einer grafischen Oberfläche auszuwerten. Zum einen können die bei der Optimierung erhaltenen Punkte in zwei- bzw. dreidimensionaler Darstellung visualisiert, zum anderen durch verschiedene Methoden, z. B. einem Entscheidungsgraph bewertet werden. Des Weiteren kann mit den in dieser Arbeit vorgestellten und implementierten Approximationsverfahren ein Ersatzmodell aufgebaut werden, mit welchem die Funktionswerte an noch nicht berechneten Punkten ermittelt werden können. Zudem wurde eine Darstellung implementiert, die den Einfluss der verschiedenen Parameter auf die Ergebnisse der einzelnen Zielfunktionen mit Hilfe des Ersatzmodells darstellt. Mit dieser Schätzung kann man in einem gewissen Maße eine Aussage darüber treffen, welche Parameter einen großen Einfluss haben und auf welche bei der Optimierung nicht so viel Wert gelegt werden muss.

A.2 Modul V-CD

Das Modul V-CD (Virtual-Chain Drive) ist ein Softwarepaket zur interaktiv unterstützten Modellen und kinetostatischen Berechnung von Kettentrieben. Dabei kann man die Konsistenz und Zulässigkeit eines Modells permanent überwachen und in Verbindung mit dem Optimierungstool der EngineeringToolbox das Kettentriebsmodell hinsichtlich verschiedener Aufgabenstellungen optimiert werden.

A.3 Modul JPPX

Das Modul JPPX (Java Post Process eXtended) dient der Visualisierung von Berechnungsergebnissen und Messdaten. Dafür stehen verschiedene Diagramm- und Kurvenarten zur Verfügung, die anschließend individuell formatiert und an das Problem angepasst werden können. Für die in dieser Arbeit erstellten Diagramme wurden kartesische Diagramme sowie 3D-Flächenkurven für die Visualisierung der Testfunktionen sowie der Approximationsergebnisse der Six-hump camel back-Funktion verwendet.

B Testfunktionen

Die folgenden Abbildungen zeigen die Originalfunktion und für jedes Approximationsverfahren und jede verwendete Verfahrenseinstellung die grafische Darstellung einer ermittelten Approximationsfunktion für die Six-hump camel back-Funktion. Dabei wurde unter den jeweils 30 ermittelten Funktionen eine solche ausgewählt, deren Gütekriterien möglichst wenig von denen des Medians abweichen.

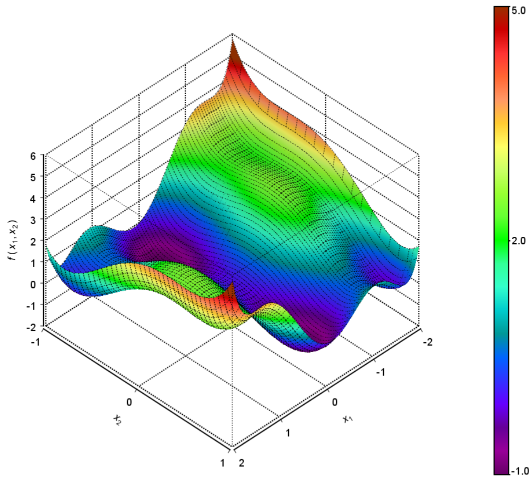

Abbildung B.1: CamelBack-Funktion

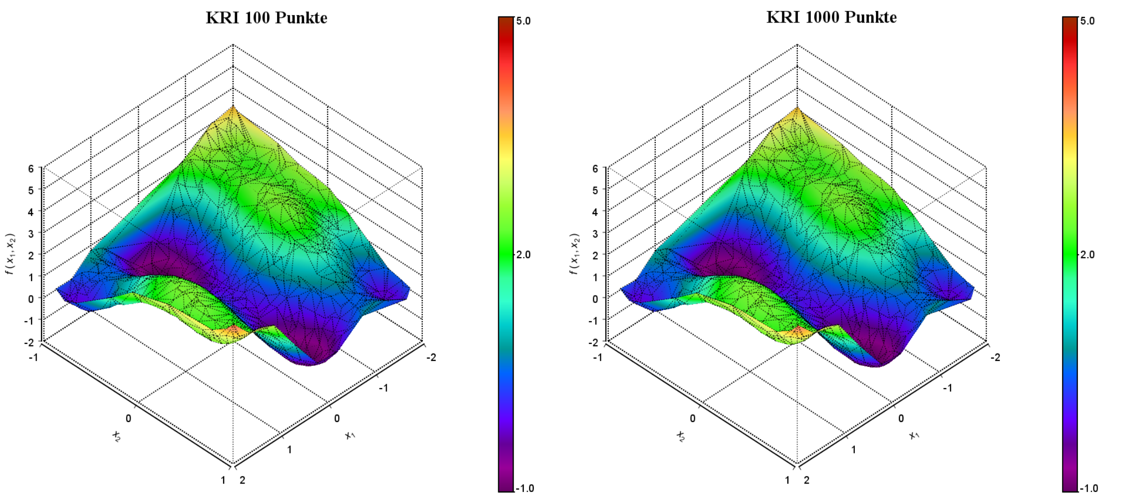

Abbildung B.2: Ergebnisse der Kriging-Approximation für die CamelBack-Funktion

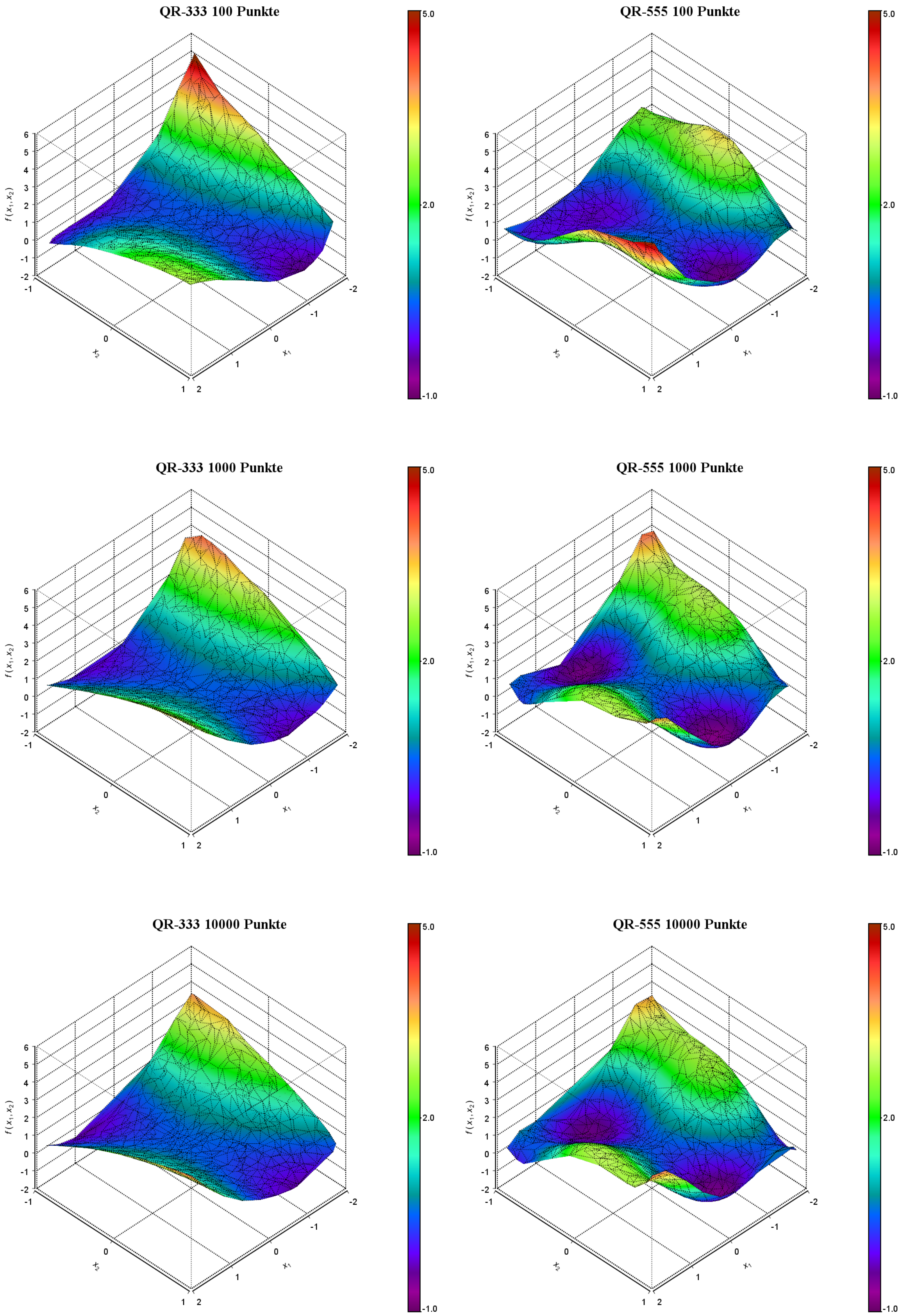

Abbildung B.3: Ergebnisse der Quasi-Regression für die CamelBack-Funktion

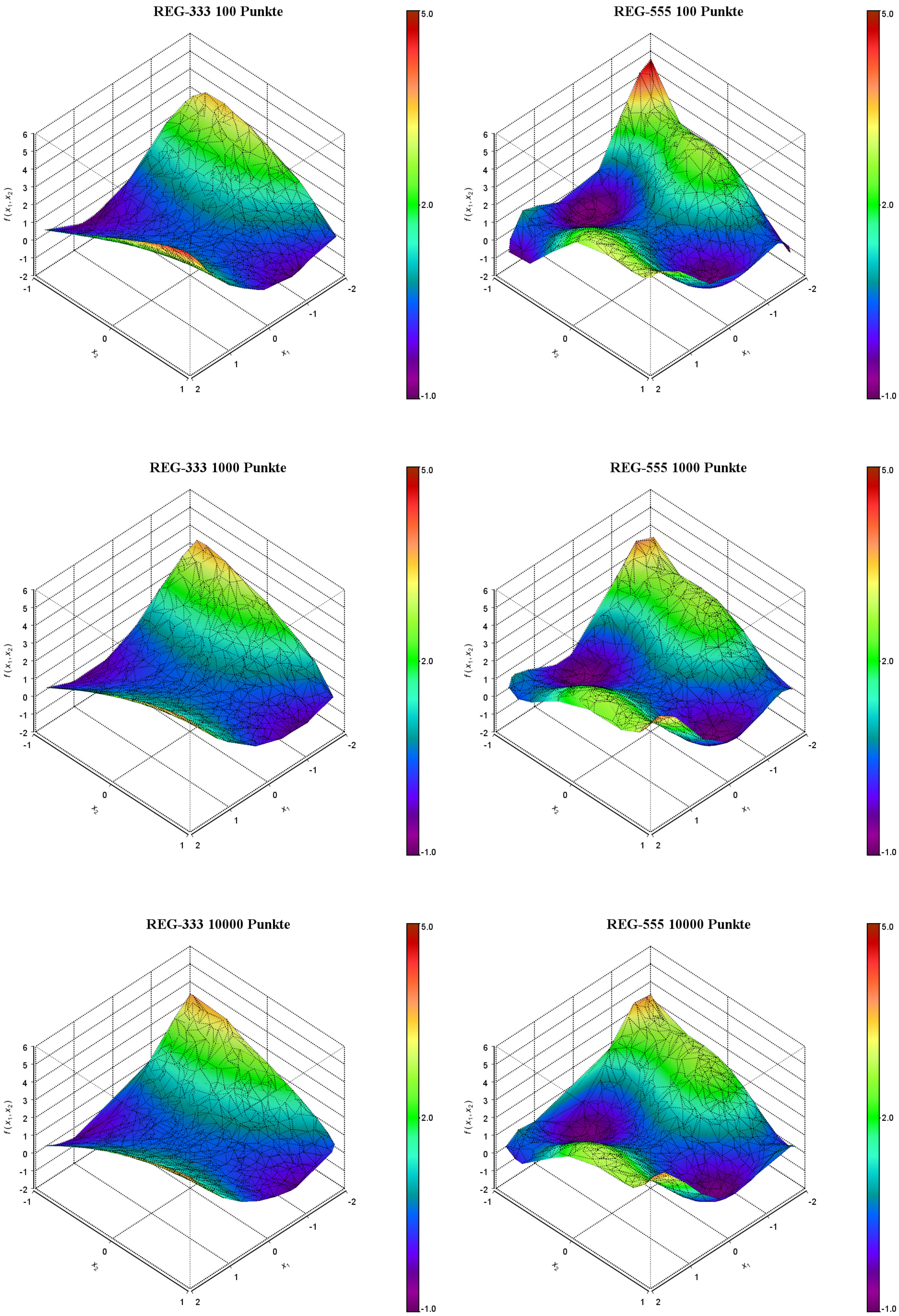

Abbildung B.4: Ergebnisse der Regression für die CamelBack-Funktion

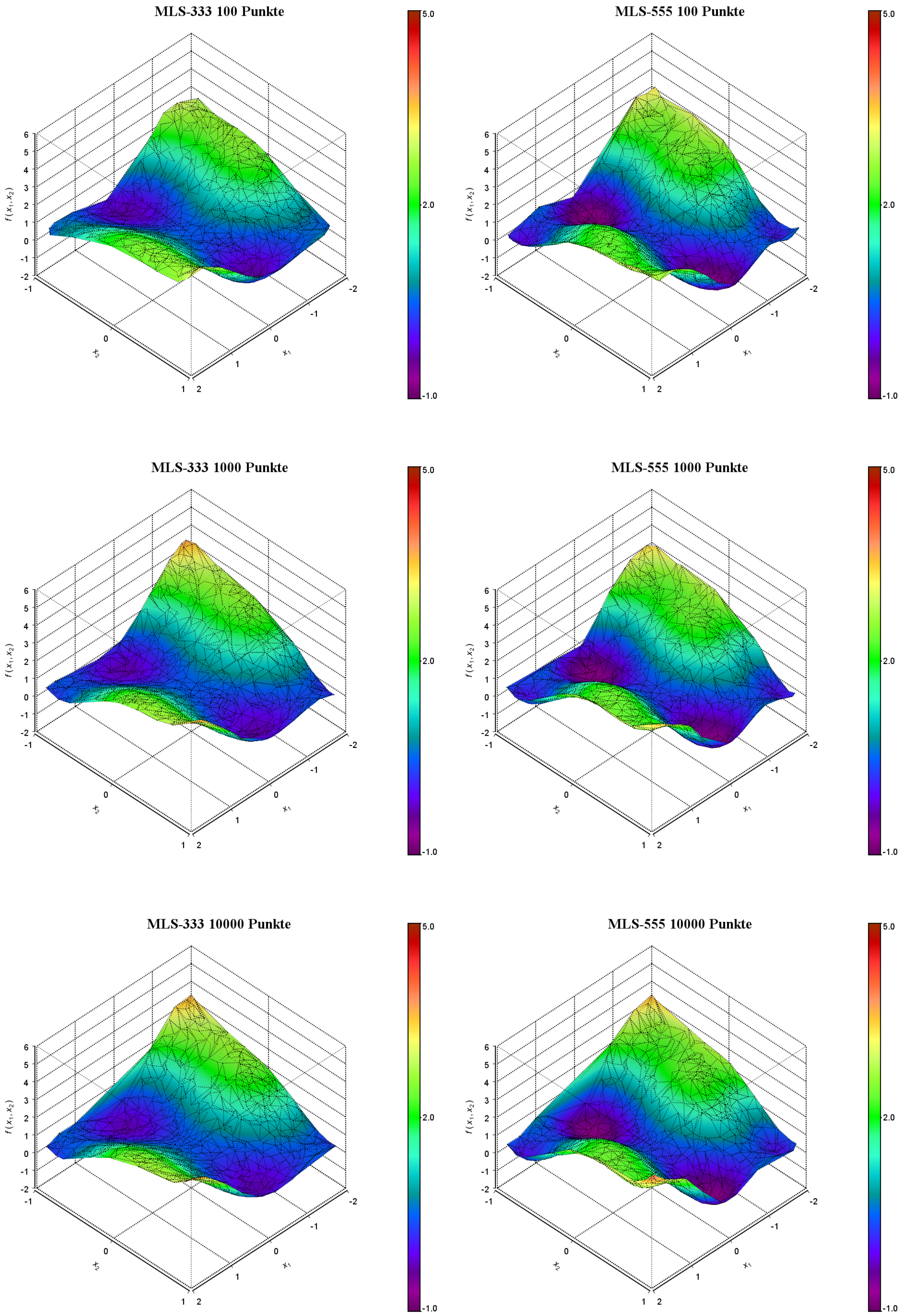

Abbildung B.5: Ergebnisse der Moving Least Squares für die CamelBack-Funktion

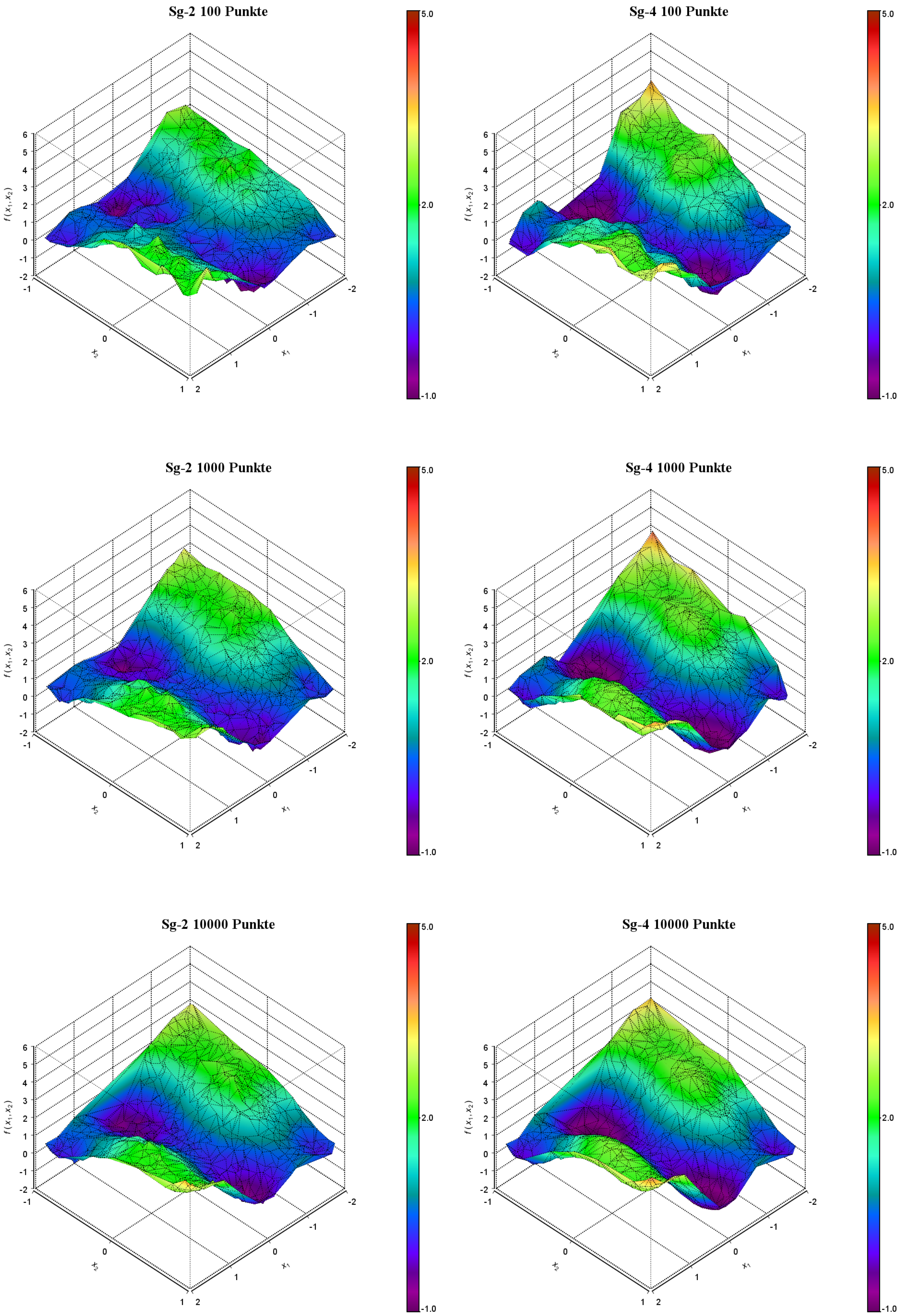

Abbildung B.6: Ergebnisse der Shepard-Interpolation für die CamelBack-Funktion

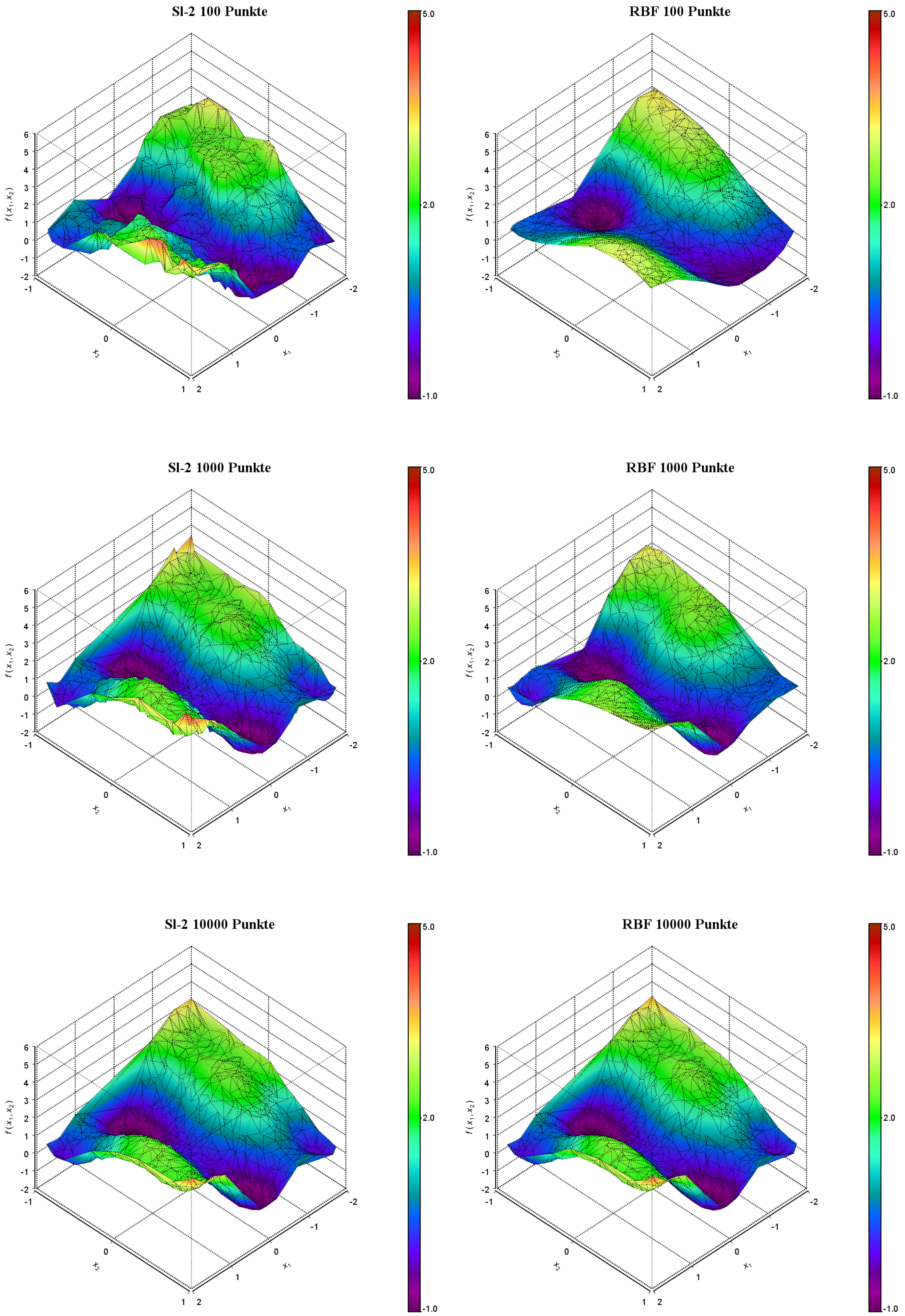

Abbildung B.7: Ergebnisse der lokalen Shepard-Interpolation und Radialen Basisfunktionen für die CamelBack-Funktion

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Chemnitz, 22. September 2010

Stefan Thämmig